# A TWO DIMENSIONAL FLUID DYNAMICS SOLVER FOR USE IN MULTIPHYSICS SIMULATIONS OF GAS COOLED REACTORS

A Thesis
Presented to
The Academic Faculty

by

Brian Lockwood

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Nuclear Engineering in the
School of Mechanical Engineering

Georgia Institute of Technology
August 2007

# A TWO DIMENSIONAL FLUID DYNAMICS SOLVER FOR USE IN MULTIPHYSICS SIMULATIONS OF GAS COOLED REACTORS

Approved by:

Dr. S. Mostafa Ghiaasiaan, Committee Chair
Mechanical Engineering
*Georgia Institute of Technology*

Dr. Cassiano de Oliveira, Advisor
Nuclear and Radiological Engineering
*Georgia Institute of Technology*

Dr. Richard C. Martineau
Multiphysics Methods Group
*Idaho National Laboratory*

Dr. W. F. G. van Rooijen
Nuclear and Radiological Engineering
*Georgia Institute of Technology*

Date Approved: 9 July 2007

# ACKNOWLEDGEMENTS

I would like to express my gratitude to the members of my reading committee for their help in preparing this thesis and their comments and feedback regarding my work. I would like to especially thank my thesis advisor, Dr. Richard Martineau, and my major professor, Dr. Cassiano de Oliveira, for their support and guidance, without which this work would not have been possible. Finally, I would like to thank my family and friends for tolerating me as I worked to finish this research.

# TABLE OF CONTENTS

# LIST OF TABLES

## LIST OF FIGURES

# SUMMARY

Currently, in the field of reactor physics, there is a drive for high fidelity, numerical simulations of reactors for the purposes of design and analysis. Since the behavior of a reactor is dependent on various physical phenomena, high fidelity simulations must be able to accurately couple these different types of physics. This is the essence of multiphysics simulations. In order to accurately simulate the thermal behavior of a reactor, the physics of neutron transport must be coupled to the fluid flow and solid phase heat conduction occurring within the reactor. This thesis develops a computational fluid dynamics solver for this purpose. The solver is based on the PCICE solution algorithm and employs cell-centered finite volumes. In addition to the fluid dynamics solver, a newly developed form of conjugate heat transfer is implemented. This implementation tightly couples the physics of solid phase heat conduction with the fluid dynamics in an efficient and consistent manner. Finally, the radiation transport code EVENT is used to provide heat generation data to the fluids solver. Using this fluids solver, several benchmark problems are analyzed and the formulation is validated.

# CHAPTER I

# INTRODUCTION

Numerical modeling has always played an important role in the development of nuclear energy. Since the beginning of the industry, computer codes have been used to gain insight into the operation and behavior of reactors. With the increase in computational speed. these codes have become increasingly sophisticated and accessible. Consequently, the use of numerical modeling has expanded and is becoming a valuable tool for the design and analysis of reactors. To aide in this use, there has been a drive to develop high fidelity modeling tools. These tools must be able to accurately reproduce the various physical phenomena present in reactors. These phenomena include such things as solid phase heat conduction, chemical reactions, neutron transport, radiative heat transfer, fluid flow and structural mechanics. These phenomena are all tightly coupled and should be simulated in a unified frame work. This unifying frame work makes up the foundation of multiphysics simulations and is an active topic of research.

The thermal behavior of a reactor is influenced mainly by the fluid flow, solid phase heat conduction and neutron transport. These three types of physics are interdependent and nonlinear. In order to achieve this coupling in a high fidelity manner, the tools of computational fluid dynamics must be combined with highly accurate computational transport tools. In this thesis, the groundwork for this combination was achieved by creating a new computational fluid dynamics solver particularly suited for use in combination with a transport solver. The computational fluid dynamics solver developed for this work consists of a finite volume formulation of the PCICE solution algorithm. The PCICE algorithm is a pressure-based temporal discretization for the conservative form of the Navier-Stokes equations. By solving the conservative form of the equations, this solver is capable of solving compressible, viscous flow over a wide range of flow regimes. Additionally, due to its treatment of pressure, the PCICE algorithm allows for larger time steps compared to traditional

temporal discretizations [15]. These characteristics give the PCICE algorithm flexibility in the time scales and flow regimes it can efficiently model, making it an ideal temporal discretization for use in multiphysics simulations.

The motivation for creating a new computational fluid dynamics solver is twofold. First, the creation of a finite volume PCICE scheme would provide an important foundation for the expanded use of algorithm. Currently, the PCICE algorithm is only implemented within a finite element formulation; however, finite volume formulations represent the dominate methodology used within the field of computational fluid dynamics. Additionally, by utilizing polyhedral cell shapes and edge based calculations, finite volume formulations are capable of higher computational efficiency and accuracy for a given number of cells [3]. The creation of a finite volume PCICE scheme would ,therefore, allow for the expanded use of the algorithm by the research community. A finite volume PCICE scheme would also provide an independent validation of the temporal behavior of the algorithm, by confirming qualities such as the temporal accuracy and stability of the scheme.

The second motivation is that the new solver can be created from the ground up with multiphysics simulations in mind. Throughout the solver, care can be taken to ensure things like mesh storage, variable assignment, and material definitions are shared between the fluid solver and the radiation transport code. Also, the ability to handle both solid and fluid phases can be included at an early stage in the solver's development, aiding in the implementation of conjugate heat transfer (CHT).

Due to its importance to reactor simulations, the solver was given the capability to solve conjugate heat transfer problems. In these problems, the heat transfer between a fluid phase and solid phase is an unknown but important quantity. They are solved by simulating the solid phase conduction and fluid flow within a single framework. Conjugate heat transfer is a relatively new field of study and previous solution methods have been based on solving the phases separately and enforcing certain continuity requirements at the interface. Although these methods work in some cases, they do not provide a strong, mathematical coupling between the heat equation of the solid and the energy equation of the fluid. In this solver, a newly developed form of conjugate heat transfer was implemented

which provides this strong coupling. Using this form of CHT, the heat transfer within the solid and fluid are treated in a uniform fashion and no boundary condition is required at the fluid/solid interface [18]. The addition of CHT capability also allows for a loose coupling between the fluid/solid solver and a neutron transport code. This loose coupling is achieved via a volumetric heat generation term based upon the fission reaction rate. Since the source term is based upon an integral quantity, a loose coupling can be made with any neutron transport code, either deterministic or Monte Carlo. For this work, the transport code EVENT was used to provide a heat source to the fluid/solid solver. The fluid/solver solver can then be used to calculate the temperature distribution throughout the domain.

In this thesis, the PCICE temporal algorithm will be outlined and its application to a finite volume spatial discretization will be detailed. In particular, a two dimensional finite volume PCICE scheme is created for the conservative form of the Navier-Stokes in two dimensional, Cartesian space. In addition to the Cartesian formulation, a finite volume formulation for the axisymmetric, Euler equations (inviscid flow) is developed and the PCICE algorithm is applied. With the finite volume formulations firmly defined, an overview of conjugate heat transfer will be given and its implementation into the finite volume formulations will be explained. Finally, the performance of the solver will be demonstrated through a series of benchmark problems. These problems range from classic problems of gas dynamics to solid phase heat conduction and conjugate heat transfer.

# CHAPTER II

# BACKGROUND

The field of fluid dynamics seeks to characterize and predict the behavior of liquids and gases in motion [20]. As such, the field is quite broad and has a number of important applications. Nearly all physical systems are influenced by fluid behavior. Characterizing this behavior has, therefore, become important in developing an understanding of these systems and the manner in which they may be utilized. Integral to the field of fluid dynamics are the Navier-Stokes equations. These equations are rooted in the principles of classical mechanics and represent the governing equations for fluid flow. This set of equations represent the conservation of mass and energy and the balance of momentum [7]. The equations themselves are typically represented by a system of coupled, nonlinear partial differential equations. Solving these equations is no small feat. In fact, few analytical solutions to the Navier-Stokes equations exist. These exact solutions employ numerous assumptions and are applicable only to very simplified problems. In general, solutions to the Navier-Stokes equations must be found numerically. The branch of fluid dynamics concerned with finding these numerical solutions is known as computational fluid dynamics (CFD). In order to solve these equations numerically, the governing equations must be discretized. Once discretized, relatively simple mathematics may be used to solve the equations.

The Navier-Stokes equations are a function of both time and space. Hence, the equations must be discretized in both time and space. The temporal discretization of the Navier-Stokes equations can take many forms. These forms range from fully explicit advancement of each variable to fully implicit formulations in which the equations are solved simultaneously. The pressure-corrected implicit continuous Eulerian (PCICE) algorithm is one such discretization. The pressure-corrected implicit continuous Eulerian (PCICE) algorithm is a pressure based method that uses a semi-implicit variable treatment to achieve favorable stability limits and strong coupling between mass and momentum [15].

The spatial discretizations used for the Navier-Stokes equations fall into three main categories. These categories are finite difference methods, finite element methods and finite volume methods. Finite difference methods seek to approximate the Navier-stokes equations by directly approximating the spatial operators in the partial differential equations with classical approximations for derivatives. Eventually, these methods gave way to finite volume and finite element methods. These methods seek to approximate the integral form of the Navier-Stokes equations. In these methods, various integral quantities are approximated [11]. Once the equations are discretized, they are solved numerically using a computer.

In this chapter, the Navier-Stokes equations will be introduced. Included in this introduction are the constitutive relations used to relate fluid properties to quantities such as stress and heat flux. Once the equations are firmly established, the PCICE temporal discretization will be derived and explained. Finally, finite volume methods will be outlined and the application of the method to the Navier-Stokes equations will be given.

## *2.1 The Governing Equations*

The governing equations for fluid flow are known as the Navier-Stokes equations. The most fundamental form is the conservative form, in which the equations represent the conservation of mass and energy and the balance of momentum. The conservative, partial differential form of the Navier-Stokes equations is given by [10]:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = s \tag{1}$$

$$\frac{\partial \rho \vec{u}}{\partial t} + \nabla \cdot (\rho \vec{u} \otimes \vec{u}) = -\nabla P + \nabla \cdot (\underline{\tau}) + \vec{b} \tag{2}$$

$$\frac{\partial \rho e_t}{\partial t} + \nabla \cdot (\rho \vec{u} h_t) = \nabla \cdot (\underline{\tau} \cdot \vec{u}) - \nabla \cdot \vec{q} + i \tag{3}$$

In these equations, $\rho$ is the density, $s$ is a mass source, $\vec{u}$ is the material velocity, $\underline{\tau}$ is the viscous stress tensor, $\vec{b}$ is a momentum source or body force per unit volume, $e_t$ is the total specific energy, $h_t$ is the total specific enthalpy, $\vec{q}$ is the conductive heat flux and $i$ is a volumetric heat generation source term. The solution variables in the above equations are known as the conserved variables. There are ultimately 3 conserved variables, the density ($\rho$), the momentum, which is a vector quantity ($\rho\vec{u}$), and the total energy ($\rho e_t$).

Corresponding to these three conserved variables are four primitive, or thermodynamic variables. These variables describe the state of the fluid, but do not specifically represent a conserved value. These four variables are the density, the material velocity ($\vec{u}$), the temperature, and the pressure. To relate these primitive variables to the conserved variables, the following observations must be made. The total energy is simply the sum of the internal energy and the kinetic energy of the fluid particle. This value is then divided by the density to yield a specific quantity. Likewise, the total enthalpy can be defined as the sum of the total energy plus the contribution of work done by pressure forces. Again, this quantity can be divided by the density to yield a per unit mass quantity. Using these definitions, the total specific energy and enthalpy are given by the following equations.

$$e_t = e + \frac{\vec{u} \cdot \vec{u}}{2} \tag{4}$$

$$h_t = \frac{\rho e_t + P}{\rho} \tag{5}$$

The specific internal energy and enthalpy can be related to the temperature through the specific heats of the fluid.

$$e = c_v T \tag{6}$$

$$h = c_p T \tag{7}$$

where $c_v$ is the specific heat at constant volume and $c_p$ is the specific heat under constant pressure.

In order to relate the pressure to the other variables in the domain, an equation of state is used to close the set of equations. This equation of state can be cast in terms of primitive variables or in terms of conversed variables. The equation of state used in this work is the ideal gas equation of state and is shown below.

$$P = f(e, \rho) \tag{8}$$

$$P = (\gamma - 1)\rho e = \rho R T \tag{9}$$

In these equations, $e$ is the specific internal energy, $\gamma$ is the ratio of specific heats (nearly a constant for Ideal Gases), $R$ is the gas constant and $T$ is the temperature of the fluid. The first form of the equation of state was used within the formulation of the scheme and within the numerical solution, while the second form was used for validation purposes due to its dependence on the primitive variables.

Finally, in order to solve the above set of equations, the viscous stress tensor and conductive heat flux must be defined. The viscous stress tensor was assumed to follow that of a newtonian fluid. The elements of this tensor can be given by:

$$\tau_{ij} = \mu(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}) - \frac{2}{3}\mu\delta_{ij}\frac{\partial u_k}{\partial x_k} \tag{10}$$

where $x_i$ represents the coordinate direction and $u_i$ represents the component of the velocity in that direction. Additionally, $\mu$ is the dynamic viscosity of the fluid and $\delta_{ij}$ is the Kronecker delta function. For the sake of clarity, the elements in the two dimensional tensor are given below.

$$\underline{\tau} = \begin{pmatrix} \frac{2}{3}\mu(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y})\hat{i} & \mu(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})\hat{i} \\ \mu(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})\hat{j} & \frac{2}{3}\mu(2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x})\hat{j} \end{pmatrix} \tag{11}$$

For the conductive heat flux, Fourier's law of conduction was assumed. This law is as follows,

$$\vec{q} = -k\nabla T \tag{12}$$

where $k$ is the thermal conductivity. In reality, this thermal conductivity can be direction dependent but in this work it was assumed to be isotropic. For fluids, the thermal conductivity can be related to the specific heat and viscosity by using the Prandtl number.

$$Pr = \frac{c_p\mu}{k} \tag{13}$$

$$k = \frac{c_p\mu}{Pr} \tag{14}$$

The above relationship was used to determine the thermal conductivity of the fluids within this work, as the Prandtl number is nearly constant over a wide range of temperatures [12].

With the governing equations established and the necessary constitutive relations defined, the equations can be discretized in both time and space.

## *2.2 Derivation of the PCICE Algorithm*

The temporal discretization used within this work is given by the PCICE algorithm. The pressure-corrected implicit continuous-fluid Eulerian (PCICE) algorithm is a temporal discretization of the conservative form of the Navier-Stokes equations. It was originally developed at the Idaho National Laboratory by Martineau and Barry and represents a significant advance in pressure-based schemes [15]. The scheme itself is valid over all flow regimes from fully compressible to the nearly incompressible [2]. Within the algorithm, the density, momentum and pressure are solved implicitly through the use of a pressure Poisson equation, which tightly couples the equations of mass conservation and momentum balance. The energy is treated semi-implicitly by applying an implicit correction to the total energy equation. The implicit variable treatment removes the acoustic component of the Courant stability limit and allows for larger maximum time steps. Additionally, this pressure Poisson equation contains an energy component derived from the equation of state. This energy component removes the need for iterations between the pressure and energy equations seen in other pressure correction schemes [15]. The PCICE algorithm is relatively new and the development of the scheme remains a topic of active research. The form used for this work is second order accurate in time for inviscid flows. For viscous flows, the scheme presented here is nearly second order accurate for high Reynolds numbers but reduces to first order for low Reynolds numbers. An overview of the original derivation found in [15] will now be given.

The target discretization for the PCICE algorithm is given below.

$$\rho^{n+1} = \rho^n - \frac{\Delta t}{2}\nabla \cdot (\rho\vec{u}^{n+1} + \rho\vec{u}^n) \tag{15}$$

$$\begin{aligned}\rho\vec{u}^{n+1} = \rho\vec{u}^n - \frac{\Delta t}{2}\nabla \cdot (\rho\vec{u} \otimes \vec{u}^p + \rho\vec{u} \otimes \vec{u}^n) + \Delta t \nabla \cdot (\underline{\tau})^n + \Delta t\, \vec{b}^n \\ - \frac{\Delta t}{2}\nabla(P^{n+1} + P^n)\end{aligned} \tag{16}$$

$$\rho e_t^{n+1} = \rho e_t^n - \frac{\Delta t}{2}\nabla \cdot (\rho\vec{u}h_t^{n+1} + \rho\vec{u}h_t^n) - \Delta t \nabla \cdot (\vec{q})^n + \Delta t \nabla \cdot (\underline{\tau} \cdot \vec{u})^n + \Delta t\, i^n \tag{17}$$

where $n$ is the solution from the previous time step and $n+1$ represents the time advanced solution. It should be noted that in the above target discretization, the mass source term is assumed to be zero. To achieve this discretization, three steps are performed. First, the variables are explicitly advanced using the convective and diffusive terms. Next, a pressure Poisson equation is constructed and solved to yield an estimate of the pressure at new time. Finally, this pressure estimate is used to update the solution variables to the new time step.

### 2.2.1 Explicit Advancement

The explicit phase of the PCICE algorithm consists of a predictor-corrector time integration. First, the new time variables are predicted using the convective fluxes. This step can be represented in compact form as:

$$\vec{U}^p = \vec{U}^n - \Delta t \nabla \cdot (\vec{F}_c^{\,n}) \tag{18}$$

$$\vec{U} = \begin{Bmatrix} \rho \\ \rho\vec{u} \\ \rho e_t \end{Bmatrix} \quad \vec{F}_c = \begin{Bmatrix} \rho\vec{u} \\ \rho\vec{u} \otimes \vec{u} \\ \rho\vec{u}h_t \end{Bmatrix}$$

where $\vec{U}^p$ represents the predicted variables and $\vec{F}_c$ represent the convective fluxes.

The solution variables are then explicitly corrected using the convective and diffusive terms. This step is represented in compact notation as:

$$\vec{U}^c = \vec{U}^n - \frac{\Delta t}{2}\nabla \cdot (\vec{F}_c^n + \vec{F}_c^p) + \Delta t \nabla \cdot \vec{F}_d^{\,n} + \vec{S}^n \tag{19}$$

$$\vec{F}_d = \left\{ \begin{array}{c} 0 \\ \underline{\tau} \\ \underline{\tau} \cdot \vec{u} - \vec{q} \end{array} \right\} \quad \vec{S} = \left\{ \begin{array}{c} 0 \\ \vec{b} \\ i \end{array} \right\}$$

where $\vec{F}_d$ represent the diffusive fluxes, $\vec{S}$ are the source terms and $\vec{U}^c$ represents the partially advanced solution variables from the corrector phase of the algorithm. As the above process shows, the convective fluxes are integrated using a two step predictor-corrector method. This method yields results that are second order accurate in time. The diffusive terms on the other hand are integrated using a one step, fully explicit time integration. For situations in which the diffusive terms dominate (low Reynolds number or solid phase conduction), the scheme will yield results which are only first order accurate in time.

### 2.2.2 The Pressure Poisson Equation

Once the solution variables are explicitly advanced using the convective and diffusive fluxes, a pressure Poisson equation is constructed in order to predict the pressure at new time and couple the equations of mass conservation and momentum balance. This equation is constructed by first taking the target discretizations for density and momentum.

$$\rho^{n+1} = \rho^n - \frac{\Delta t}{2}\nabla \cdot (\rho\vec{u}^{n+1} + \rho\vec{u}^n) \tag{20}$$

$$\begin{aligned} \rho\vec{u}^{n+1} = \rho\vec{u}^n - \frac{\Delta t}{2}\nabla \cdot (\rho\vec{u} \otimes \vec{u}^p + \rho\vec{u} \otimes \vec{u}^n) - \frac{\Delta t}{2}\nabla(P^{n+1} + P^n) + \\ \Delta t \nabla \cdot (\underline{\tau}^n) + \Delta t\, \vec{b}^n \end{aligned} \tag{21}$$

The target discretization for the momentum can now be rewritten in terms of the explicitly advanced momentum, $\rho\vec{u}^c$. Once the equation is rewritten, the momentum at time level $n$ is added to the equation. The result is given by:

$$\rho\vec{u}^n + \rho\vec{u}^{n+1} = \rho\vec{u}^n + \rho\vec{u}^c - \frac{\Delta t}{2}\nabla(P^{n+1} + P^n) \tag{22}$$

The right hand side of the equation is now identical to the convective flux seen in the target discretization for the density. By substituting equation 22 into equation 20, an elliptical equation in terms of pressure can be found.

$$\rho^{n+1} - \rho^n = \frac{\Delta t^2}{4}\nabla\cdot\nabla(P^{n+1} + P^n) - \frac{\Delta t}{2}\nabla\cdot(\rho\vec{u}^c + \rho\vec{u}^n) \tag{23}$$

At this point, the equations of momentum and mass are tightly coupled. Simply put, the pressure at new time will yield a momentum field that also satisfies mass conservation at the new time level. To solve this equation however, the density at new time level must be related to the pressure at time level $n+1$. To achieve this end, the equation of state is used. By taking the total derivative of the equation of state, the following relationship can be established.

$$P = f(\rho, e) \tag{24}$$

$$\delta P = \frac{\partial f}{\partial \rho}\delta\rho + \frac{\partial f}{\partial e}\delta e \tag{25}$$

This relationship can then be used to relate the change of density with respect to time to the change in pressure with respect to time.

$$\rho^{n+1} - \rho^n = \frac{1}{\frac{\partial f}{\partial \rho}^*}(P^{n+1} - P^n) - \frac{\frac{\partial f}{\partial e}^*}{\frac{\partial f}{\partial \rho}^*}(e^{n+1} - e^n) \tag{26}$$

where the superscript $*$ indicates that the quantity is evaluated at some time between $n$ and $n+1$. Since the specific internal energy at new time is not known, the above equation can be approximated using the explicitly advanced internal energy. For simplicity, the derivatives of the equation of state are also evaluated using the explicitly advanced variables.

$$\rho^{n+1} - \rho^n \approx \frac{1}{\frac{\partial f}{\partial \rho}^c}(\hat{P}^{n+1} - P^n) - \frac{\frac{\partial f}{\partial e}^c}{\frac{\partial f}{\partial \rho}^c}(e^c - e^n) \tag{27}$$

In the above equation, $\hat{P}^{n+1}$ is now an estimate of the pressure at the new time level, since it only approximately satisfies the equation of state. Substituting this approximation into the coupled mass-momentum equation, the final form of the pressure Poisson equation is found.

$$\frac{\hat{P}^{n+1} - P^n}{\left(\frac{\partial f}{\partial \rho}\right)^c} - \frac{\Delta t^2}{4}\nabla\cdot\nabla(\hat{P}^{n+1} - P^n) = \frac{\left(\frac{\partial f}{\partial e}\right)^c}{\left(\frac{\partial f}{\partial \rho}\right)^c}(e^c - e^n) - \frac{\Delta t}{2}\nabla\cdot(\rho\vec{u}^c + \rho\vec{u}^n) + \frac{\Delta t^2}{2}\nabla\cdot\nabla(P^n) \tag{28}$$

This equation can then be solved using any method valid for elliptic equations.

### 2.2.3 Pressure-Correction

To determine the correction step needed for each variable, the equation for the explicit advancement is compared to the target discretization. The pressure-correction step is then given by the difference between the two equations. For the momentum variable, this comparison yields a correction step in which the variables are simply updated using the pressure forces.

$$\rho\vec{u}^{n+1} = \rho\vec{u}^c - \frac{\Delta t}{2}\nabla(\hat{P}^{n+1} - P^n) \tag{29}$$

To determine the density at time $n+1$, the momentum at new time is used to correct the convective flux term of the mass equation. In order to yield the target discretization, the predicted flux must be subtracted from this correction. The density correction step is given by:

$$\rho^{n+1} = \rho^c - \frac{\Delta t}{2}\nabla(\rho\vec{u}^{n+1} - \rho\vec{u}^p) \tag{30}$$

Finally, to find the total energy at time $n+1$, the new momentum is used to correct the convective flux in the energy equation, just as with the density. However, In the case of the energy, the total specific enthalpy at time $n+1$ is required to make this correction. Since the total enthalpy at new time depends on the total energy, this value must be estimated. To estimate this value, the new time pressure estimate and density are used in conjunction with the explicity advanced total energy.

$$h_t^{n+1} = \frac{\rho e_t^c + \hat{P}^{n+1}}{\rho^{n+1}} \tag{31}$$

Using this approximation, the correction step for the energy becomes:

$$\rho e_t^{n+1} = \rho e_t^c - \frac{\Delta t}{2}\nabla(\rho\vec{u}h_t^{n+1} - \rho\vec{u}h_t^p) \tag{32}$$

Finally, with the conserved variables at new time found, the thermodynamic variables are calculated using the new time values. This includes a recalculation of pressure at new time. This recalculation uses the equation of state. In this manner, the new time pressure is guaranteed to satisfy the equation of state exactly. The difference between the estimated new time pressure and the actual new time pressure is in practice quite small. However, if the pressure is not recalculated after each time step, the small error quickly builds and becomes a source of instability within the solution [15]. With the PCICE algorithm introduced, it is prudent to discuss the spatial discretization used within this work.

### *2.3 Finite Volume Formulations*

Finite volume methods are a class of spatial discretization that attempt to approximate the integral form of the Navier-Stokes equations. These methods were initially an extension of finite difference methods. Over time, they have evolved into a distinct set of numerical techniques. When compared to finite difference discretizations, there are several advantages to finite volume methods First, complex geometries can be analyzed without transforming the equations into a computational space. Additionally, structured grids and coordinate transformations are not required [11]. Finally, cell shapes other than quadrilaterals can be analyzed, giving rise to more efficient and accurate solution schemes [3]. The application of the finite volume method to the Navier-Stokes equations will be developed below.

In order to form a finite volume formulation of Navier-Stokes, the partial differential form of the equations presented above must be transformed into integral equations. To aid in this derivation, the compact form of the equations are defined [10].

$$\frac{\partial\vec{U}}{\partial t} + \nabla\cdot\vec{F} = \nabla\cdot\vec{F}_v + \vec{S} \tag{33}$$

$$\vec{U} = \begin{Bmatrix} \rho \\ \rho\vec{u} \\ \rho e_t \end{Bmatrix} \qquad \vec{F} = \begin{Bmatrix} \rho\vec{u} \\ \rho\vec{u} \otimes \vec{u} + P \\ \rho\vec{u}h_t \end{Bmatrix}$$

$$\vec{F_v} = \begin{Bmatrix} 0 \\ \underline{\tau} \\ \underline{\tau} \cdot \vec{u} + k\nabla T \end{Bmatrix} \quad \vec{S} = \begin{Bmatrix} 0 \\ \vec{b} \\ i \end{Bmatrix}$$

This equation (or set of equations) can now be integrated over an arbitrary volume in space.

$$\int_{V_a(t)} \frac{\partial \vec{U}}{\partial t} dV + \int_{V_a(t)} \nabla \cdot \vec{F} dV = \int_{V_a(t)} \nabla \cdot \vec{F_v} dV + \int_{V_a(t)} \vec{S} dV \tag{34}$$

Using the divergence theorem, the flux terms can be converted into integrals over the surface area of the volume.

$$\int_{V_a(t)} \frac{\partial \vec{U}}{\partial t} dV + \int_{A_a(t)} \vec{F} \cdot d\vec{A} = \int_{A_a(t)} \vec{F_v} d\vec{A} + \int_{V_a(t)} \vec{S} dV \tag{35}$$

To yield the final finite volume integral equation, the arbitrary volume will be assumed to be fixed in time and equal to one of the geometrical cells in the discretized domain. Although the assumption of fixed cells in time is not necessary, it simplifies the equations. With the volume unchanging, the partial derivative with respect to time can be brought outside the volume integral and the Eulerian form of the finite volume equations are found.

$$\frac{\partial}{\partial t} \int_{V_c} \vec{U} dV + \int_{A_c} \vec{F} \cdot d\vec{A} = \int_{A_c} \vec{F_v} \cdot d\vec{A} + \int_{V_c} \vec{S} dV \tag{36}$$

where $V_c$ and $A_c$ refer to the volume and surface area of the cell. The above equation is valid for all Eulerian finite volume methods in one, two, or three dimensions. The equation will now be simplified in accordance with the scope of this work. First, the remaining volume integrals can be represented as the average of the quantity over the cell. Second, assuming the arbitrary volume is a polyhedral, the integral over the surface area can be approximated as a sum over the faces of the cell.

$$V_c \frac{d}{dt}\vec{U}_i + \sum_k \vec{F}_k \cdot \vec{n}_k A_k = \sum_k \vec{F}_{v,k} \cdot \vec{n}_k A_k + V_c \vec{S}_i \tag{37}$$

where $\vec{U}_i$ is the average of $\vec{U}$ over the cell volume, $\vec{S}_i$ is the average source over the cell. The vectors $\vec{F}_k$ and $\vec{F}_{v,k}$ are the average flux vectors over the $k^{th}$ face. Finally, $\vec{n}_k$ is the unit normal and $A_k$ is the surface area of the $k^{th}$ face. Often these two terms are combined to form $\vec{A}_k$.

The finite volume equation is now reduced to two dimensions using the following definitions [3].

$$\vec{F}_k = \vec{f}_k \hat{i} + \vec{g}_k \hat{j} \qquad \vec{F}_{v,k} = \vec{f}_{v,k} \hat{i} + \vec{g}_{v,k} \hat{j} \tag{38}$$

$$\vec{A}_k = \vec{n}_k A_k = dy_k \hat{i} - dx_k \hat{j} \tag{39}$$

where $dy_k$ and $dx_k$ refer to the change in the y and x coordinates seen across the face. In practice, this value is found by subtracting the X and Y coordinates of the two nodes defining the cell face. By numbering the nodes and subtracting them in a consistent manner (such as counter-clockwise), the proper direction of the surface normal can guaranteed. In addition to defining the surface normal, the flux vectors have been decomposed into their components in the X and Y directions. Using these definitions, the inner products along each face can be written explicitly.

$$V_i \frac{d}{dt}\vec{U}_i + \sum_k \vec{f} dy_k - \vec{g} dx_k = \sum_k \vec{f}_v dy_k - \vec{g}_v dx_k + V_i \vec{S}_i \tag{40}$$

For clarity, these equations will be written for each of the four conserved variables present in two dimensions.

$$V_i \frac{d}{dt}\rho_i + \sum_k \rho u dy_k - \rho v dx_k = 0 \tag{41}$$

$$V_i \frac{d}{dt}\rho u_i + \sum_k (\rho u^2 + P) dy_k - \rho uv dx_k = \sum_k \tau_{xx} dy_k - \tau_{xy} dx_k + V_i b_{x,i} \tag{42}$$

$$V_i \frac{d}{dt}\rho v_i + \sum_k \rho vu dy_k - (\rho v^2 + P) dx_k = \sum_k \tau_{yx} dy_k - \tau_{yy} dx_k + V_i b_{y,i} \tag{43}$$

$$V_i \frac{d}{dt}\rho e_{t,i} + \sum_k \rho u h_t dy_k - \rho v h_t dx_k = \sum_k (\tau_{xx} u + \tau_{xy} v) dy_k - (\tau_{yx} u + \tau_{yy} v) dx_k \tag{44}$$

$$- \sum_k q_x dy_k - q_y dx_k + V_i i_i \tag{45}$$

The above equations represent the spatially discretized navier stokes equations. In the two dimensional case, the cell volume is taken to mean the cell area. The term volume is used to avoid confusion with the surface area of the cell, which in two dimensions becomes the perimeter. These finite volume equations are valid for any type of Eulerian finite volume scheme in two dimensions. In the next section, these equations will be specialized further. The PCICE algorithm will then be used to discretize these equations in time.

# CHAPTER III

# FINITE VOLUME BASED PCICE SCHEME (PCICE-FVM)

With the introduction of the Navier-Stokes equations and the outlining of the temporal and spatial discretizations completed, the development of a new computational fluid dynamics solver can be explained. For this solver, the finite volume method was applied to the PCICE temporal discretization. This temporal discretization is independent of space and should be equally valid for any spatial discretization. Currently, the algorithm is implemented within a finite element formulation. In order to provide independent validation of the PCICE temporal discretization and further its use within the CFD community, a finite volume PCICE scheme (PCICE-FVM) was created. The development of this scheme represents the bulk of this thesis and is the subject of the following chapter. First, the manner in which cell averaged and face centered quantities are determined will be defined. With these definitions, a finite volume PCICE scheme in two dimensional, Cartesian space is rigorously derived. Following this derivation, an axisymmetric formulation for inviscid flow will be outlined. Finally, the application of artificial viscosity, stability of the scheme and implementation of boundary conditions will be explained.

## *3.1 Determination of Cell Averaged and Face Centered Quantities*

Before deriving the equations for a finite volume PCICE scheme, certain engineering choices regarding the finite volume formulation must be established. These choices relate to the definitions of cell averaged and face centered quantities and they affect such things as accuracy, ease of implementation, and stability. The choices made for the PCICE-FVM scheme will be explained below.

The first freedom allowed in a finite volume formulation is the definition of the cell averaged quantities ($\vec{U}_i$). The choice of definition determines the location at which variables are defined and the nature of the equations to be solved. For this work, the variables evaluated at the geometrical center of a cell are taken to be the cell averaged quantity. This

definition classifies the formulation as cell centered. Additionally, since all of the variables are defined at the cell centers, the formulation is said to use a coallocated grid [4]. With the choice of a cell-centered scheme, the finite volume equations now represent a series of coupled ordinary differential equations.

With the choice of cell centered finite volumes, the value of the flux at the cell face must be interpolated using cell centered quantities. The order of accuracy used in this interpolation determines the overall spatial accuracy of the scheme. For this work, linear interpolation was used. To perform this linear interpolation, the flux was evaluated at the centers of the cells adjacent to the face. It was then assumed that the value of the flux varied linearly along the line connecting the two cell centers. If this line intersects the face center, as is the case with an orthogonal grid, the interpolation will yield a second order accurate approximation of the flux surface integral. As the line connecting the cell centers intersects further from the midpoint of the face, the order of accuracy will decrease. The interpolation can be represented by [7]:

$$f_k = \frac{|\vec{r}_k - \vec{r}_i|}{|\vec{r}_j - \vec{r}_i|} f_i + \frac{|\vec{r}_k - \vec{r}_j|}{|\vec{r}_j - \vec{r}_i|} f_j \tag{46}$$

where $f_i$ and $f_j$ are the flux values evaluated at the centers of the two cells adjacent to face $k$. The vectors $\vec{r}_i$, $\vec{r}_j$ and $\vec{r}_k$ are the locations of the two cell centers and the face center, respectively. On a uniform, orthogonal grid, this interpolation reduces to taking the arithmetic mean of the fluxes.

$$f_k = \frac{1}{2}(f_i + f_j) \tag{47}$$

This interpolation is used to determine all face centered quantities unless specified otherwise.

In some cases, the partial derivative of a variable with respect to each coordinate direction is needed at the cell center. This is the case with the viscous stress terms, in which the partial derivatives of the velocity are needed. Additionally, the pressure gradient is required in the momentum equation. To determine the cell centered partial derivative, the approximation is derived as follows [5].

$$\int_{A_i} \frac{\partial \Phi}{\partial x}_i dxdy = \int_{L_i} \Phi dy \tag{48}$$

$$\int_{L_i} \Phi dy = \sum_k \Phi_k dy_k \tag{49}$$

$$\frac{\partial \Phi}{\partial x}_i = \frac{1}{A_i} \sum_k \Phi_k dy_k \tag{50}$$

where $\Phi_k$ is the face centered quantity and is computed using the linear interpolation outlined above.

This same process can be done similarly for the other derivatives. Partial derivatives with respect to y are approximated by:

$$\int_{A_i} \frac{\partial \Phi}{\partial y}_i dxdy = \int_{L_i} \Phi dx \tag{51}$$

$$\int_{L_i} \Phi dx = -\sum_k \Phi_k dx_k \tag{52}$$

$$\frac{\partial \Phi}{\partial y}_i = -\frac{1}{A_i} \sum_k \Phi_k dx_k \tag{53}$$

To approximate the Laplacian of a quantity, as seen in the pressure Poisson equation, the method described above could be carried out twice and the Laplacian at the center of a cell could be calculated. Although this method would require fewer assumptions about the orientation of the grid, it would be computationally expensive and yield a larger zone of dependence for each cell [7]. The cell centered Laplacian of a variable can be directly approximated as follows. First, the divergence theorem is utilized to create a surface integral.

$$\int_{V_i} \nabla \cdot \nabla \Phi = \int_{A_i} \nabla \Phi \cdot d\vec{A} \tag{54}$$

$$\int_{A_i} \nabla \Phi \cdot d\vec{A} = \int_{A_i} \frac{\partial \Phi}{\partial n} |dA| \tag{55}$$

where $\frac{\partial \Phi}{\partial n}$ is the derivative of $\Phi$ in the direction normal to the surface. Approximately, this surface integral can be represented as:

$$\int_{A_i} \frac{\partial \Phi}{\partial n} |dA| = \sum_k \frac{\partial \Phi}{\partial n}_k |dA|_k \tag{56}$$

If the normal to the surface is assumed to be parallel to the line connecting the cell centers adjacent to face $k$, then the derivative with respect to this normal direction can be approximated directly using only the cells adjacent to face $k$. This approximation is given by [7]:

$$\frac{\partial \Phi}{\partial n}_k = \frac{\Phi_j - \Phi_i}{|\vec{r}_j - \vec{r}_i|} \tag{57}$$

Hence, the final form of the approximation is the following.

$$\int_{V_i} \nabla \cdot \nabla \Phi dV = \sum_k \frac{\Phi_j - \Phi_i}{|\vec{r}_j - \vec{r}_i|} |dA_k| \tag{58}$$

For a structured, uniform grid, the above approximation reduces to the standard central differencing approximation of the Laplacian.

## *3.2 PCICE Algorithm with Finite Volumes*

Using the definitions from the previous section, the finite volume method was applied to the equations of the PCICE temporal discretization. For the purposes of this work, the PCICE algorithm was separated into four steps. These steps were explicit convection, explicit diffusion, the pressure Poisson step and the pressure correction step. The development of a finite volume formulation for each of these steps is outlined below.

### 3.2.1 Explicit Convection

The first step of the PCICE algorithm is a two step time integration of the convective flux vectors. This two step integration takes the form of a predictor-corrector. The predictor step of this integration is again represented by:

$$\vec{U}^P = \vec{U}^n - \Delta t \nabla \cdot \vec{F}_c^n \tag{59}$$

This equation can now be integrated over the volume of a cell.

$$\int_{V_i} \vec{U}^p = \int_{V_i} \vec{U}^n - \Delta t \int_{A_i} \vec{F}_c^n \cdot \vec{dA} \tag{60}$$

The equation can be simplified using the methods outlined in the previous section. First, $U_i$ is defined as the solution vector at the center of the cell. The surface integral is then approximated as the sum over the sides of the volume. Next, the problem is restricted to two dimensions by replacing the cell volume with the cell area and by writing the convective flux in terms of its components in the X and Y coordinate directions, $f_c$ and $g_c$.

$$\vec{f_c} = \left\{ \begin{array}{c} \rho u \\ \rho u^2 \\ \rho uv \\ \rho u h_t \end{array} \right\} \quad \vec{g_c} = \left\{ \begin{array}{c} \rho v \\ \rho uv \\ \rho v^2 \\ \rho v h_t \end{array} \right\}$$

With these simplifications, the predictor step of the explicit advancement becomes the following:

$$A_i \vec{U}_i^p = A_i \vec{U}_i^n - \Delta t \sum_k f_c^n dy_k - g_c^n dx_k \tag{61}$$

Similarly, the corrector step can be represented as follows:

$$A_i \vec{U}_i^{c'} = A_i \vec{U}_i^n - \frac{\Delta t}{2} \sum_k (f_c^n + f_c^p) dy_k - (g_c^n + g_c^p) dx_k \tag{62}$$

### 3.2.2 Explicit Diffusion Step

Once the solution variables are explicitly convected, the contributions due to diffusive fluxes and source terms are added. This addition is treated as a separate step simply for clarity and convenience. In the numerical implementation of the formulation, these additional terms are treated using a separate loop through the variables in order to ease in solving inviscid problems. Since addition is the only operator being split, there is no error encountered in this separate treatment. The diffusive step is represented by the following equation.

$$\vec{U}^c = \vec{U}^{c'} + \Delta t \nabla \cdot \vec{F}_d + \Delta t \vec{S} \tag{63}$$

where $\vec{U}^{c'}$ is the solution after the explicit convection step. The diffusive flux is now broken into a diffusive flux caused by viscous stress terms and a diffusive flux caused by

conduction, $F_q$. These vectors can be represented as:

$$\vec{F}_v = \begin{Bmatrix} 0 \\ \tau \\ \tau \cdot \vec{u} \end{Bmatrix} \quad \vec{F}_q = \begin{Bmatrix} 0 \\ 0 \\ -\vec{q} \end{Bmatrix}$$

The diffusion step is then integrated over the cell volume and the divergence theorem is applied. The result of this manipulation is the following:

$$\int_{V_i} \vec{U}^c = \int_{V_i} \vec{U}^{c'} + \Delta t \int_{A_i} \vec{F}_v^n \cdot d\vec{A} + \Delta t \int_{A_i} \vec{F}_q \cdot d\vec{A} + \Delta t \int_{V_i} \vec{S} dV \tag{64}$$

At this point, the problem is restricted to two dimensions, causing the volume integral to become an area integral and the surface integral to become a line integral. Defining $f_v$ as the viscous flux in the x direction and $g_v$ as the viscous flux in the y direction, the viscous flux can be written as a sum over the sides of the cell and the inner product with the surface normal can be explicitly evaluated.

$$A_i \vec{U}_i^c = A_i \vec{U}_i^{c'} + \Delta t \sum_k f_v^n dy_k - g_v^n dx_k + \Delta t \sum_k \vec{F}_{q,k} \cdot \vec{dl}_k + \Delta t A_i \vec{S}_i \tag{65}$$

where $f_v$ and $g_v$ are given by the following:

$$\vec{f}_v = \begin{Bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xx} u + \tau_{xy} v \end{Bmatrix} \quad \vec{g}_v = \begin{Bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ \tau_{xy} u + \tau_{yy} v \end{Bmatrix}$$

The viscous flux is treated in a manner identical to the convective fluxes. The flux is evaluated at the cell centers adjacent to the face and the face centered value is determined using linear interpolation. The viscous stress at the cell center is given by:

$$\tau_{xx,i} = \frac{2\mu_i}{3}(2\frac{\partial u}{\partial x}_i - \frac{\partial v}{\partial y}_i) \tag{66}$$

$$\tau_{xy,i} = \mu_i(\frac{\partial u}{\partial y}_i + \frac{\partial v}{\partial x}_i) \tag{67}$$

$$\tau_{yy,i} = \frac{2\mu_i}{3}(2\frac{\partial v}{\partial y}_i - \frac{\partial u}{\partial x}_i) \tag{68}$$

where the partial derivatives of the velocity are determined using the approximation in Equation 48. The heat flux on the other hand was approximated in a more direct manner. Instead of determining cell centered derivatives and interpolating, the heat flux normal to a face was approximated directly using the two cells adjacent to the face. The treatment of this flux will be treated in the following chapter as it is formulated with conjugate heat transfer in mind. For now, the thermal flux normal to a face will be denoted as $F_{q,n}$. The inner product is then written as the heat flux normal to the face multiplied by the length of the face. The final diffusive correction step is given by:

$$A_i\vec{U}_i^c = A_i\vec{U}_i^{c'} + \Delta t\sum_k f_v^n dy_k - g_v^n dx_k + \Delta t\sum_k F_{q,n}dl_k + \Delta t A_i\vec{S}_i \tag{69}$$

With the solution variables explicitly advanced, the pressure Poisson equation is formed and the new time pressure is estimated in an implicit manner.

### 3.2.3 Pressure Poisson Equation

The Pressure Poisson equation is used to predict the pressure at time $n+1$. This implicit pressure is then used to update the solution variables. By leaving out the contribution of pressure in the explicit steps, the acoustic component of the CFL limit is removed, increasing it to the material Courant limit. The general pressure Poisson equation is given by:

$$\frac{\hat{P}^{n+1}-P^n}{\left(\frac{\partial f}{\partial \rho}\right)^c} - \frac{\Delta t^2}{4}\nabla\cdot\nabla(\hat{P}^{n+1}-P^n) = \frac{\left(\frac{\partial f}{\partial e}\right)^c}{\left(\frac{\partial f}{\partial \rho}\right)^c}(e^c-e^n) - \frac{\Delta t}{2}\nabla\cdot(\rho\vec{u}^c+\rho\vec{u}^n) + \frac{\Delta t^2}{2}\nabla\cdot\nabla(P^n) \tag{70}$$

where $\hat{P}^{n+1}$ is the predicted new time pressure, $e$ is the specific internal energy and $f$ is the equation of state. Defining $\delta P$ as the difference between the new time pressure prediction and the old time pressure, the above equation can be simplified to the following:

$$\frac{\delta P}{\left(\frac{\partial f}{\partial \rho}\right)^c} - \frac{\Delta t^2}{4}\nabla\cdot\nabla(\delta P) = \frac{\left(\frac{\partial f}{\partial e}\right)^c}{\left(\frac{\partial f}{\partial \rho}\right)^c}(e^c-e^n) - \frac{\Delta t}{2}\nabla\cdot(\rho\vec{u}^c+\rho\vec{u}^n) + \frac{\Delta t^2}{2}\nabla\cdot\nabla(P^n) \tag{71}$$

For this work, the following form of the ideal gas equation of state was used.

$$f(\rho, e) = (\gamma - 1)\rho e \tag{72}$$

With this equation of state, the partial derivatives with respect to each variable are given by:

$$\frac{\partial f}{\partial \rho} = (\gamma - 1)e \tag{73}$$

$$\frac{\partial f}{\partial e} = (\gamma - 1)\rho \tag{74}$$

Using this equation of state and the above derivatives, the Pressure Poisson equation becomes:

$$\frac{\delta P}{(\gamma - 1)e^c} - \frac{\Delta t^2}{4}\nabla \cdot \nabla(\delta P) = \frac{\rho^c}{e^c}(e^c - e^n) - \frac{\Delta t}{2}\nabla \cdot (\rho \vec{u}^c + \rho \vec{u}^n) + \frac{\Delta t^2}{2}\nabla \cdot \nabla(P^n) \tag{75}$$

To create a finite volume form of the equation, it is integrated over the cell volume and the necessary approximations are applied . Here as before, the divergence theorem is used to yield surface integrals.

$$\begin{aligned}\int_{V_i} \frac{\delta P}{(\gamma - 1)e^c} dV - \frac{\Delta t^2}{4}\int_{A_i} \nabla(\delta P) \cdot d\vec{A} = \int_{V_i} \frac{\rho^c}{e^c}(e^c - e^n) dV \\ - \frac{\Delta t}{2}\int_{A_i} (\rho \vec{u}^c + \rho \vec{u}^n) \cdot d\vec{A} + \frac{\Delta t^2}{2}\int_{A_i} \nabla(P^n) \cdot d\vec{A}\end{aligned} \tag{76}$$

The equation is reduced to two dimensions by replacing the integral over the volume by an integration over the area and the surface integral by a line integral. Next, the integrals over the Area can be replaced by the cell centered value multiplied by the area of the cell. Finally, as before, the line integral can be approximated by a sum over the faces of the integrand evaluated at the face center. Using these approximations, the pressure Poisson equation becomes.

$$\begin{aligned}\frac{A_i \delta P_i}{(\gamma - 1)e_i^c} - \frac{\Delta t^2}{4}\sum_k \nabla(\delta P) \cdot \vec{dl}_k = A_i \frac{\rho_i^c}{e_i^c}(e_i^c - e_i^n) \\ - \frac{\Delta t}{2}\sum_k [(\rho u^c + \rho u^n) dy_k - (\rho v^c + \rho v^n) dx_k] + \frac{\Delta t^2}{2}\sum_k \nabla P^n \cdot \vec{dl}_k\end{aligned} \tag{77}$$

In the above equation, the mass flux term has been approximated in the same manner as in the explicit step. In the above equation, the gradient terms have only been abstractly represented. To approximate these terms, the approximation given in Equation 57 is utilized. Using this approximation, the equation becomes:

$$\begin{aligned}\frac{A_i \delta P_i}{(\gamma-1)e_i^c} - \frac{\Delta t^2}{4}\sum_k \frac{\delta P_j - \delta P_i}{|r_j - r_i|}|dl_k| = A_i \frac{\rho_i^c}{e_i^c}(e_i^c - e_i^n) \\ - \frac{\Delta t}{2}\sum_k [(\rho u^c + \rho u^n)dy_k - (\rho v^c + \rho v^n)dx_k] + \frac{\Delta t^2}{2}\sum_k \frac{\delta P_j - \delta P_i}{|r_j - r_i|}|dl_k|\end{aligned} \tag{78}$$

Using the above equation as a template, a set of linear equations for $\delta P$ was constructed. From inspection, one can see some of the qualities of the matrix representing this set of equations. First, it can be seen that $\delta P_i$ depends only on the change in pressure in the neighboring cells. This small footprint is caused by the approximation used for the gradient of pressure at a face. It should be noted that this small footprint is at the expense of fully unstructured mesh capability (meaning that restrictions are placed on the non-orthogonality allowed in the mesh) [7]. The small footprint also suggests that the matrix will be sparse and appropriate accommodation in terms of computational storage of the matrix and solving of the matrix should be taken. The second observation is that the dependence of $\delta P_i$ on the neighboring values is due to the gradient and mass flux terms. This fact suggests that the matrix will be symmetric, allowing for the use of more advanced numerical solvers of the linear equations.

Additionally, it should be noted that one could have applied the finite volume formulation to the target discretization and then derived the pressure Poisson equation. This approach was not chosen for two reasons. First, the PCICE algorithm is a temporal discretization independent of spatial discretization. By discretizing the equations in space before deriving the steps of the algorithm, the space and time discretizations would become inexorably linked. Second, in the case of the pressure Poisson equation, discretizing the equations in space before constructed the pressure Poisson equation would have led to a large computational footprint for each cell. By creating the pressure Poisson equation first

and then integrating over the cell volume, the terms within the equation, such as the Laplacian of the pressure, can be approximated directly, instead of merely applying a differencing to the cell centered gradients.

### 3.2.4 Pressure-Correction Step

Once the pressure at new time is predicted, the solution variables are updated according to this new pressure information. This is achieved by three separate loops over the variables. These update steps are listed below in the order in which they are performed.

$$\rho\vec{u}^{n+1} = \rho\vec{u}^{c} - \frac{\Delta t}{2}\nabla(\hat{P}^{n+1} + P^{n}) \tag{79}$$

$$\rho^{n+1} = \rho^{c} - \frac{\Delta t}{2}\nabla\cdot(\rho\vec{u}^{n+1} - \rho\vec{u}^{p}) \tag{80}$$

$$\rho e_t^{n+1} = \rho e_t^{c} - \frac{\Delta t}{2}\nabla\cdot(\rho\vec{u}h_t^{n+1} - \rho\vec{u}h_t^{p}) \tag{81}$$

To give the finite volume form, the same process carried out on the explicit convection equations was performed on the update equations. This process will be shown for the momentum equation. In two dimensions, there are two momentum balance equations. The update step for these two equations is given by:

$$\rho u^{n+1} = \rho u^{c} - \frac{\Delta t}{2}\frac{\partial(\hat{P}^{n+1} + P^{n})}{\partial x} \tag{82}$$

$$\rho v^{n+1} = \rho v^{c} - \frac{\Delta t}{2}\frac{\partial(\hat{P}^{n+1} + P^{n})}{\partial y} \tag{83}$$

These equations are now integrated over the area of the cell. (Note: since the problem is already restricted to 2-D, the integration is performed over the Area, not the volume as before)

$$\int_{A_i}\rho u^{n+1}dA = \int_{A_i}\rho u^{c}dA - \frac{\Delta t}{2}\int_{A_i}\frac{\partial}{\partial x}(\hat{P}^{n+1} + P^{n})dA \tag{84}$$

$$\int_{A_i}\rho v^{n+1}dA = \int_{A_i}\rho v^{c}dA - \frac{\Delta t}{2}\int_{A_i}\frac{\partial}{\partial y}(\hat{P}^{n+1} + P^{n})dA \tag{85}$$

At this point, the equation could be cast in a form such that divergence theorem can be used upon the pressure term. However, for reasons which will become clear once the axisymmetric form of the scheme is outlined, the pressure term is treated as a source term [7]. Hence, in the above equation, all of the area integrals are replaced by the cell centered value multiplied by the area of the cell. The equation now becomes:

$$A_i \rho u_i^{n+1} = A_i \rho u_i^c - \frac{\Delta t}{2} A_i \frac{\partial (\hat{P}^{n+1} + P^n)}{\partial x}_i \tag{86}$$

$$A_i \rho v_i^{n+1} = A_i \rho v_i^c - \frac{\Delta t}{2} A_i \frac{\partial (\hat{P}^{n+1} + P^n)}{\partial y}_i \tag{87}$$

In the above equations, the cell centered partial derivative of pressure is approximated using Equation 48. Using this approximation, the pressure-correction equations for the momentum is given by:

$$A_i \rho u_i^{n+1} = A_i \rho u_i^c - \frac{\Delta t}{2} \sum_k (\hat{P}^{n+1} + P^n) dy_k \tag{88}$$

$$A_i \rho v_i^{n+1} = A_i \rho v_i^c + \frac{\Delta t}{2} \sum_k (\hat{P}^{n+1} + P^n) dx_k \tag{89}$$

The above equation is identical to the equation derived using the divergence theorem on the gradient of the pressure in Cartesian space. For the axisymmetric formulation, there is an important difference which will be explained in a later section.

The pressure-correction step for the density and energy are nearly identical to the explicit convection of these variables. Since the development of the finite volume formulation for this type of equation is firmly established at this point, only the final equations will be presented. For the density, the update equation is given by:

$$A_i \rho_i^{n+1} = A_i \rho_i^c - \frac{\Delta t}{2} \sum_k [(\rho u^{n+1} - \rho u^p) dy_k - (\rho v^{n+1} - \rho v^p) dx_k] \tag{90}$$

For the energy, a similar equation is used to bring the variable to the new time level.

$$A_i \rho e_{t,i}^{n+1} = A_i \rho e_{t,i}^c - \frac{\Delta t}{2} \sum_k [(\rho u h_t^{n+1} - \rho u h_t^p) dy_k - (\rho v h_t^{n+1} - \rho v h_t^p) dx_k] \tag{91}$$

where $h_t^{n+1}$ is given by:

$$h_{t,i}^{n+1} = \frac{\rho e_{t,i}^c + \hat{P}_i^{n+1}}{\rho_i^{n+1}} \tag{92}$$

Once the conserved variables are advanced to the new time level, the primitive variables, including pressure, are computed for use in the next time step.

### *3.3 Axisymmetric Formulation*

In addition to the finite volume formulation developed for Cartesian coordinates, a formulation was created for the PCICE algorithm in axisymmetric coordinates. As was the case for Cartesian coordinates, the axisymmetric PCICE algorithm has been previously implemented using only finite elements [14]. To simplify the finite volume formulation, only the equations for inviscid flow (Euler equations) were solved. Axisymmetric flow occurs in cylindrical geometry (r,$\theta$,z), when the variables do not vary in the $\theta$ direction. This two-dimensional approximation is one of the most common and is valid for many engineering systems. The flow equations for inviscid, axisymmetric flow are given below [9].

$$\frac{\partial(r\rho)}{\partial t} + \frac{\partial(r\rho v_z)}{\partial z} + \frac{\partial(r\rho v_r)}{\partial r} = 0 \tag{93}$$

$$\frac{\partial(r\rho v_z)}{\partial t} + \frac{\partial(r\rho v_z v_z)}{\partial z} + \frac{\partial(r\rho v_z v_r)}{\partial r} = -r\frac{\partial P}{\partial z} \tag{94}$$

$$\frac{\partial(r\rho v_r)}{\partial t} + \frac{\partial(r\rho v_r v_z)}{\partial z} + \frac{\partial(r\rho v_r v_r)}{\partial r} = -r\frac{\partial P}{\partial r} \tag{95}$$

$$\frac{\partial(r\rho e_t)}{\partial t} + \frac{\partial(r\rho v_z h_t)}{\partial z} + \frac{\partial(r\rho v_r h_t)}{\partial r} = 0 \tag{96}$$

With the assumption that the variables are independent of $\theta$, the problem is reduced to two dimensions. Using $r$ and $z$ as the position variables, the coordinate space is now orthogonal. The divergence operator and gradient operator in this space can now be represented by [14]:

$$\nabla \cdot (f\hat{z} + g\hat{r}) = \frac{\partial f}{\partial z} + \frac{\partial g}{\partial r} \tag{97}$$

$$\nabla\Phi = \frac{\partial \Phi}{\partial z} + \frac{\partial \Phi}{\partial r} \tag{98}$$

Using this notation, the flow equations can be rewritten as: (Notice the Momentum equations have been rewritten in vector form)

$$\frac{\partial(r\rho)}{\partial t} + \nabla \cdot r\rho\vec{v} = 0 \tag{99}$$

$$\frac{\partial(r\rho\vec{v})}{\partial t} + \nabla \cdot r(\rho\vec{v} \otimes \vec{v}) = -r\nabla P \tag{100}$$

$$\frac{\partial(r\rho e_t)}{\partial t} + \nabla \cdot r\rho\vec{v}h_t = 0 \tag{101}$$

The PCICE temporal discretization can now be applied to this set of equations. A full derivation of the axisymmetric PCICE algorithm can be found in Reference [14]. This derivation is nearly identical to the Cartesian derivation and will not be given. Based on the derivation in Reference [14], the equations for each step of the algorithm and the finite volume treatment of each of these steps will be explained.

### 3.3.1 Axisymmetric Explicit Convection

The discretized equations for the axisymmetric explicit convection step are given by the following two step time integration.

$$r\vec{U}^p = r\vec{U}^n - \Delta t \nabla \cdot r\vec{F}_c^n \tag{102}$$

$$r\vec{U}^c = r\vec{U}^n - \frac{\Delta t}{2}\nabla \cdot r(\vec{F}_c^p + \vec{F}_c^n) \tag{103}$$

Integration over the Area of the cell and applying the divergence theorem yields the following finite volume equations:

$$A_i r_i \vec{U}_i^p = A_i r_i \vec{U}_i^n - \Delta t \sum_k f^n dr_k - g^n dz_k \tag{104}$$

$$A_i r_i \vec{U}_i^c = A_i r_i \vec{U}_i^n - \frac{\Delta t}{2} \sum_k (f^n + f^p) dr_k - (g^p + g^n) dz_k \tag{105}$$

where the vector $\vec{U}$ has the same value as before, $r_i$ is the radius of the cell center and the $f$ and $g$ now take on the following values:

$$\vec{f}_c = \begin{Bmatrix} r\rho v_z \\ r\rho v_z^2 \\ r\rho v_z v_z \\ r\rho v_z h_t \end{Bmatrix} \quad \vec{g}_c = \begin{Bmatrix} r\rho v_r \\ r\rho v_r v_z \\ r\rho v_r^2 \\ r\rho v_r h_t \end{Bmatrix}$$

The above flux vectors are interpolated to the cell face in the same manner as the Cartesian case. It should be noted that the radius is included in this interpolation. For clarity, the interpolation for the mass flux is shown below. Note that the vector $\vec{s}$ is now used to denote the position vector to avoid confusion with the scalar $r$ which represents the radius.

$$(r\rho v_z)_k = \frac{|\vec{s}_k - \vec{s}_i|}{|\vec{s}_j - \vec{s}_i|}[r_i(\rho v_z)_i] + \frac{|\vec{s}_k - \vec{s}_j|}{|\vec{s}_j - \vec{s}_i|}[r_j(\rho v_z)_j] \tag{106}$$

$$(r\rho v_r)_k = \frac{|\vec{s}_k - \vec{s}_i|}{|\vec{s}_j - \vec{s}_i|}[r_i(\rho v_r)_i] + \frac{|\vec{s}_k - \vec{s}_j|}{|\vec{s}_j - \vec{s}_i|}[r_j(\rho v_r)_j] \tag{107}$$

Since the axisymmetric formulation only seeks to solve the Euler equations, there is no need for an explicit diffusion step. Therefore, after the variables are explicitly convected, the pressure Poisson equation is constructed and solved.

### 3.3.2 Axisymmetric Pressure Poisson Equation

The axisymmetric form of the pressure Poisson equation with the ideal gas equation of state and written in terms of $\delta P$ is given by:

$$\frac{r\delta P}{(\gamma-1)e^c} - \frac{\Delta t^2}{4}\nabla \cdot r\nabla(\delta P) = r\frac{\rho^c}{e^c}(e^c - e^n) - \frac{\Delta t}{2}\nabla \cdot r(\rho\vec{v}^c + \rho\vec{v}^n) + \frac{\Delta t^2}{2}\nabla \cdot r\nabla(P^n) \tag{108}$$

When this equation is integrated over the cell area and the divergence theorem is utilized, the equation becomes:

$$\begin{aligned}\frac{A_i r_i \delta P_i}{(\gamma-1)e_i^c} - \frac{\Delta t^2}{4}\sum_k r_k \nabla(\delta P)\cdot\vec{dl}_k &= A_i r_i \frac{\rho_i^c}{e_i^c}(e_i^c - e_i^n) \\ &- \frac{\Delta t}{2}\sum_k [(r\rho u^c + r\rho u^n)dy_k - (r\rho v^c + r\rho v^n)dx_k] + \frac{\Delta t^2}{2}\sum_k r_k \nabla P^n \cdot \vec{dl}_k\end{aligned} \tag{109}$$

where $r_i$ denotes the radius of the cell center, $r_k$ is the radius of the face center, and the other variables are defined as before. Finally, by utilizing the approximation given in Equation 57, the final form of the axisymmetric Pressure Poisson equation is found.

$$\frac{A_i r_i \delta P_i}{(\gamma-1)e_i^c} - \frac{\Delta t^2}{4}\sum_k r_k \frac{\delta P_j - \delta P_i}{|\vec{s}_j - \vec{s}_i|}|dl_k| = A_i r_i \frac{\rho_i^c}{e_i^c}(e_i^c - e_i^n) - \frac{\Delta t}{2}\sum_k [(r\rho u^c + r\rho u^n)dy_k - (r\rho v^c + r\rho v^n)dx_k] + \frac{\Delta t^2}{2}\sum_k r_k \frac{\delta P_j - \delta P_i}{|\vec{s}_j - \vec{s}_i|}|dl_k| \tag{110}$$

### 3.3.3 Axisymmetric Pressure-Correction Step

The axisymmetric Pressure-Correction equations are given below for each of the solution variables.

$$r\rho\vec{u}^{n+1} = r\rho\vec{u}^c - \frac{\Delta t}{2} r\nabla(\hat{P}^{n+1} + P^n) \tag{111}$$

$$r\rho^{n+1} = r\rho^c - \frac{\Delta t}{2}\nabla \cdot r(\rho\vec{u}^{n+1} - \rho\vec{u}^p) \tag{112}$$

$$r\rho e_t^{n+1} = r\rho e_t^c - \frac{\Delta t}{2}\nabla \cdot r(\rho\vec{u}h_t^{n+1} - \rho\vec{u}h_t^p) \tag{113}$$

The momentum pressure-correction equation is now integrated over the cell area to yield the finite volume form of the equation. Recall that the pressure gradient is treated as a source term in the equation. This treatment causes the radius to appear outside of the pressure gradient approximation. Therefore, the finite volume equation becomes:

$$A_i r_i \rho\vec{u}_i^{n+1} = A_i r_i \rho\vec{u}_i^c - \frac{\Delta t}{2} A_i r_i (\nabla[\hat{P}^{n+1} + P^n)]_i \tag{114}$$

Again using the approximation given in Equation 48, the pressure correction equation for each component of the momentum can be written as follows:

$$A_i r_i \rho v_{z,i}^{n+1} = A_i r_i \rho v_{z,i}^c - \frac{\Delta t}{2} r_i \sum_k (\hat{P}^{n+1} + P^n)dr_k \tag{115}$$

$$A_i r_i \rho v_{r,i}^{n+1} = A_i r_i \rho v_{r,i}^c + \frac{\Delta t}{2} r_i \sum_k (\hat{P}^{n+1} + P^n)dz_k \tag{116}$$

The derivation of the other pressure correction equations is straight forward and will be omitted. The final form of these equations are given below. For the density, the equation is given by:

$$A_i r_i \rho_i^{n+1} = A_i r_i \rho_i^c - \frac{\Delta t}{2} \sum_k (r\rho v_z^{n+1} - r\rho v_z^p) dr_k - (r\rho v_r^{n+1} - r\rho v_r^p) dz_k \tag{117}$$

Finally, the axisymmetric energy pressure correction equation is given by:

$$A_i r_i \rho e_{t,i}^{n+1} = A_i r_i \rho e_{t,i}^c - \frac{\Delta t}{2} \sum_k (r\rho v_z h_t^{n+1} - r\rho v_z h_t^p) dr_k - (r\rho v_r h_t^{n+1} - r\rho v_r h_t^p) dz_k \tag{118}$$

where $h_t^{n+1}$ is defined the same as in the Cartesian formulation (Equation 92).

### 3.3.4 Comparison to Cartesian Formulation

With the axisymmetric formulation outlined, it is prudent to make a comparison between the Cartesian and axisymmetric formulations. Additionally, it should be explained how a single computer code can be used to solve both kinds of problems. The first observation that can be made is that if $r = 1$, the $z$ axis is aligned with the $x$ axis and the $r$ axis is aligned with the $y$ axis, the axisymmetric formulation reduces to the Cartesian formulation. This can be seen by setting $r = 1$, $dz$ to $dx$ and $dr$ to $dy$ in any of the previously outlined equations. For example, with these substitutions, the axisymmetric momentum pressure-correction equation becomes:

$$A_i(1) \rho v_{z,i}^{n+1} = A_i(1) \rho v_{z,i}^c - \frac{\Delta t}{2}(1) \sum_k (\hat{P}^{n+1} + P^n) dy_k \tag{119}$$

$$A_i(1) \rho v_{r,i}^{n+1} = A_i(1) \rho v_{r,i}^c + \frac{\Delta t}{2}(1) \sum_k (\hat{P}^{n+1} + P^n) dx_k \tag{120}$$

The above equations are identical to the momentum pressure correction equations in the Cartesian formulation. Thus, if a computer code is written to solve the axisymmetric form of the equations, it can be easily adjusted to solve Cartesian problems by simply making the afore mentioned substitutions. If the opposite case is true and a computer code has

been written to solve the Cartesian form of the equations, the adjustment of only a few terms will allow the code to solve axisymmetric problems.

When the discretized axisymmetric Equations are examined, one observes that the cell Area always appears multiplied by the cell centered radius. Although it would be possible to modify the code to add an additional term in each equation, a more simple solution is preposed by Peric. By redefining the Area to include the radius of the cell center, the number of modifications to the code is greatly reduced. This definition is given by:

$$\tilde{A}_i = \frac{1}{6}\sum_{i=1}^{N_v}(z_{i-1} - z_i)(r_{i-1}^2 + r_i^2 + r_i r_{i-1}) \tag{121}$$

where $N_v$ is the total number of vertices belonging to the cell. These vertices are numbered in a counter-clockwise fashion with $i = 0$ corresponding to $i = N_v$ [7]. Aside from the altered definition of the Area, the flux vectors must be modified to include the radius. This addition is relatively simple and merely requires the cell centered radius to be multiplied by the cell centered flux vector before interpolating the value to the face. Finally, in the pressure Poisson equation and the momentum pressure-correction equations, the pressure gradient must be multiplied by the radius. In the case of the momentum pressure-correction, the cell centered radius is used as this term is treated as a source. In the pressure Poisson equation, since the radius is inside of a divergence, the radius of the face center must be used. By making these relatively minor changes, a computer code can be used to solve both Cartesian and axisymmetric problems. In fact, the ease in which this may be achieved was one of the large motivations in exploring both coordinate systems.

### *3.4 Jameson's Artificial Dissipation*

Given the choice of linear interpolation, nonphysical oscillations are introduced into the system. These oscillations are most pronounced around shocks and discontinuities. For Inviscid problems, they can cause the solution to become unstable [21]. To remove these oscillations and stabilize the calculations, Jameson's form of artificial dissipation (artificial viscosity) was used. This artificial dissipation works by applying a diffusive flux to the solution which is proportional to the second and fourth derivatives of the solution vector.

The Laplacian term is used to remove the large scale oscillations encountered around shocks. The biharmonic term is used to remove small scale oscillations present as the system tends toward steady state. Without this biharmonic term, the system may not converge to a steady state solution. This artificial dissipation is treated as an additional flux term [13].

### 3.4.1 Definition

With the addition of artificial diffusion, the governing equations in compact form for inviscid flow can be represented as:

$$\frac{\partial U}{\partial t} + \nabla \cdot \vec{F} - \nabla \cdot \vec{F_{av}} = 0 \tag{122}$$

For a finite volume, this equation can be discretized in space to yield [13]:

$$V_i \frac{\partial U_i}{\partial t} + Q_i(U) - D_i(U) = 0 \tag{123}$$

where:

$$Q_i(U) = \sum_k F(\vec{U})_k \cdot \vec{A}_k \tag{124}$$

$$D_i(U) = \sum_k D(\vec{U})_k \tag{125}$$

For Jameson's dissipation, the diffusive flux for a face can be represented by the following [13]:

$$D_k = \lambda_k(\varepsilon_k^{(2)} \nabla U - \varepsilon_k^{(4)} \nabla(\nabla^2 U)) \tag{126}$$

Instead of determining the exact gradient in the solution and the Laplacian of the solution, an unscaled difference is used as it is computationally inexpensive and has minimal effect on the artifical dissipation. For the gradient in the solution, the difference between the two cells adjacent to the face are used. For the gradient in the Laplacian of the solution, the difference between the Laplacian evaluated at the cell centers of the adjacent cells are used. Here again, the Laplacian is approximated by an unscaled difference. The Laplacian in the solution variable $U$ for cell $i$ is given by [1]:

$$(\nabla^2 U)_i \approx \sum_k U_j - U_i \tag{127}$$

Using these approximations, the diffusive flux at a face becomes:

$$D_k = \lambda_k(\varepsilon_k^{(2)}(U_j - U_i) - \varepsilon_k^{(4)}(\nabla^2 U_j - \nabla^2 U_i)) \tag{128}$$

The constants in this equation are adapted to match the flow. For areas in which the flow is constant or varies linear, no dissipation should be applied. Near shocks, the second order term should become dominate and the fourth order term should be set to zero due to its tendency to induce overshoots. To meet these requirements, the constants are defined as [13]:

$$\varepsilon_k^{(2)} = \kappa^{(2)} max(\nu_i, \nu_j) \tag{129}$$

$$\varepsilon_k^{(4)} = max(0, \kappa^{(4)} - \varepsilon^2) \tag{130}$$

where $v_i$ is an approximation of the Laplacian in the pressure and $\kappa^{(2)}$ and $\kappa^{(4)}$ are user defined constants. The Laplacian in pressure is approximated by a pressure switch. This pressure switch is zero in regions of constant or linear pressure and is a maximum around relative extrema in pressure. The pressure switch is defined as [21]:

$$\nu_i = \sum_k \frac{|P_j - P_i|}{(P_j + P_i)} \tag{131}$$

Finally, $\lambda_k$ is used to give the term the proper dimensionality and appropriate scaling. Originally, this constant was defined in two dimensions as [13]:

$$\lambda_k = \frac{A_i + A_j}{2\Delta t} \tag{132}$$

where $A_i$ is the area of the $i^{th}$ cell and $A_j$ is the area of the $j^{th}$ cell. Later, this constant was redefined in order to remove the dependence on the time step.

$$\lambda_k = \frac{\lambda_i + \lambda_j}{2} \tag{133}$$

where:

$$\lambda_i = \sum_k (u dy_k - v dx_k) + c_k \sqrt{dx_k^2 + dy_k^2} \tag{134}$$

In this equation, $u_k$ and $v_k$ are the X and Y components of the velocity at the face and $c_k$ is the speed of sound at the face [19]. Intuitively, this constant can be seen as the integral over the surface of the maximum eigenvalue of the Euler equations [1]. This definition removes the dependence on time step, thus making the amount of dissipation added to the flow independent of the size of the time step. It is this definition that is implemented in the current work.

Using the latter definition of $\lambda_k$, typical smoothing constants are [19]:

$$\kappa^{(2)} = \frac{1}{2} \qquad \kappa^{(4)} = \frac{1}{128} \tag{135}$$

For axisymmetric flows, the definitions outlined above are kept exactly the same, with the exception of $\lambda_i$. This constant is redefined for axisymmetric flow as:

$$\tilde{\lambda}_i = \sum_k r_k (u dy_k - v dx_k) + r_k c_k \sqrt{dx_k^2 + dy_k^2} \tag{136}$$

where $r_k$ is the radius of the face center. Aside from this alternative definition, the calculation and application of artificial dissipation in axisymmetric flow is the same as the Cartesian case.

### 3.4.2 Application of Artificial Dissipation

Nonphysical oscillations can be introduced during the evaluation of any first order operator. Therefore, artificial dissipation is applied whenever one of these operators is evaluated. These operators include the divergence operator and the gradient operator. Therefore, artificial diffusion is applied during each step of the PCICE-FVM scheme, except for the explicit diffusion step. The specifics of each application are outlined below for the Cartesian formulation. This application is identical for the axisymmetric formulation. At the beginning of a time step, the constant $\lambda$ and pressure switch, $\nu$, are calculated using the solution variables from the previous time step. These two values are only computed once per time step and the same value of these constants is used for each application throughout

the time step. The artificial dissipation is first applied during the explicit convection step of the PCICE-FVM scheme. This step with artificial dissipation is represented by:

$$A_i\vec{U}^p = A_i\vec{U}^n - \Delta t Q(U)^n + \Delta t D(U)^n \tag{137}$$

$$A_i\vec{U}^c = A_i\vec{U}^n - \frac{\Delta t}{2}[Q(U)^n + Q(U)^p] + \frac{\Delta t}{2}[D(U)^n + D(U)^p] \tag{138}$$

where $Q(\vec{U})^n$ and $D(\vec{U})^n$ denote the convective flux and artificial diffusive flux evaluated using the variables at time $n$. Notice that the dissipation term is added during both the predictor and corrector step. While the time level of the solution used to evaluate the diffusive flux changes, the constants within the diffusion operator are not recomputed. Since the explicit diffusion step does not involve any first order operators, no artificial dissipation is applied in this step.

In the forcing function of the pressure Poisson equation, the divergence of the mass flux is evaluated. Since the approximation of this operator can induce oscillations, artificial dissipation is applied to this value as well. The pressure Poisson equation with the dissipation component is given by:

$$\begin{aligned}\frac{A_i\delta P_i}{(\gamma-1)e_i^c} - \frac{\Delta t^2}{4}\sum_k \frac{\delta P_j - \delta P_i}{|r_j - r_i|}|dl_k| &= A_i\frac{\rho_i^c}{e_i^c}(e_i^c - e_i^n) \\ - \frac{\Delta t}{2}[Q(\rho)^c + Q(\rho)^n] + \frac{\Delta t}{2}[D(\rho)^c + D(\rho)^n] &+ \frac{\Delta t^2}{2}\sum_k \frac{\delta P_j - \delta P_i}{|r_j - r_i|}|dl_k|\end{aligned} \tag{139}$$

where $Q(\rho)$ denotes the integral over the cell boundary of the mass flux at the specified time level. Artificial dissipation is not applied to the pressure terms in the equation because these terms ultimately reduce to second order operators.

Finally, in the pressure correction equations, artificial dissipation is again applied. First, the momentum equations are solved.

$$A_i\rho u^{n+1} = A_i\rho u^c - \frac{\Delta t}{2}[S_x(P)^{n+1} + S_x(P)^n] + \frac{\Delta t}{2}[D(\rho u)^c + D(\rho u)^n] \tag{140}$$

$$A_i\rho v^{n+1} = A_i\rho u^c - \frac{\Delta t}{2}[S_y(P)^{n+1} + S_y(P)^n] + \frac{\Delta t}{2}[D(\rho v)^c + D(\rho v)^n] \tag{141}$$

where $S$ is defined as:

$$S_x(P) = \sum_k P_k dy_k \tag{142}$$

$$S_y(P) = -\sum_k P_k dx_k \tag{143}$$

The pressure-correction equations for the density and total energy with the inclusion of artificial dissipation are now given by:

$$A_i\rho^{n+1} = A_i\rho^c - \frac{\Delta t}{2}[Q(\rho)^{n+1} - Q(\rho)^p] + \frac{\Delta t}{2}[D(\rho)^c - D(\rho)^n] \tag{144}$$

$$A_i\rho e_t^{n+1} = A_i\rho e_t^c - \frac{\Delta t}{2}[Q(\rho e_t)^{n+1} - Q(\rho e_t)^p] + \frac{\Delta t}{2}[D(\rho e_t)^c - D(\rho e_t)^n] \tag{145}$$

As the above equations illustrate, the amount of dissipation added in this step is the difference between the dissipation at the pressure-correction step and at the predictor step. In this way, the amount of dissipation applied to these variables is reduced relative to the momentum. However, it can be argued that the momentum variable is the variable that requires the most dissipation. The use of linear interpolation in pressure-based schemes (central differencing using the finite differencing terminology) can in many cases lead to spatial decoupling of the solution. This decoupling occurs because the mass flux for a given cell and pressure force are determined by different approximations. Because of these differing approximations, two separate solutions can develop in the domain and the "checkerboarding" effect is observed. When this "checkerboarding" occurs, the pressure and momentum variables become decoupled [7]. By applying artificial dissipation to both the convective and pressure terms of the momentum equation, these terms are given a dependence on the cell for which they are evaluated and decoupling is prevented.

The method in which artificial dissipation is applied is flexible and the afore outlined method represents merely one of many possible methods. Other variants would include treating the artificial dissipation as a spatial smoother and applying it a singe time at the end of each time step. Additionally, the dissipative term could be frozen within a time step in the same manner as $\lambda$ and the pressure switch [13]. Both of these methods would result

in a different amount of dissipation to be applied to the solution but would yield roughly the same answers. The advantage of a given strategy over another must be determined based on the overall effect on the solution and the computational effort required to implement the strategy.

### *3.5 Stability of the PCICE-FVM Scheme*

One of the main advantages of the PCICE algorithm is the removal of the acoustic component from the Courant stability criterion. In this section, stability of the finite volume PCICE scheme will be explained. The stability criterion for a one step, fully explicit formulation of the Euler equations is written as:

$$CFL = \frac{(|\vec{u}| + c)\Delta t}{\Delta s} \leq 1 \tag{146}$$

where $\Delta s$ is a representative mesh spacing (typically the minimum cell spacing). This stability requirement simply states that information in the domain must not propagate more than one cell in a time step. The fastest rate at which information is propagated is given by the maximum characteristic of the Euler equations $(u + c)$ [10]. By treating time integration by multiple steps, such as a Runge Kutta scheme, stability can be achieved for CFL numbers greater than 1 [13]. The acoustic component of this stability limit occurs because in an explicit scheme, pressure information is transmitted through the domain in a similar manner as density and energy. As such, the time scale of the simulation must be small enough to capture this propagation. By treating the pressure implicitly in the form of an elliptical equation, the pressure information can be propagated instantly throughout the domain with no stability limit. Therefore, the only stability limit for the scheme is the rate at which information can be convected through the domain. For the PCICE algorithm in inviscid flow, the maximum time step can be given by:

$$\Delta t \leq \frac{C\Delta s}{2|\vec{u}|} \tag{147}$$

where $C$ is a user supplied safety factor [15]. For the Finite volume formulation of the PCICE algorithm, a conservative safety factor of 0.2 appears to be valid for a large number

of problems. The maximum safety factor used within this work is 0.8, which corresponds to high speed flow ($Ma \approx 1$). It is likely that in this case, the time step is sufficiently limited by the velocity, decreasing the impact of the safety factor. The safety factor required in the finite volume formulation appears to be lower than the safety factor used in the finite element formulation, which is approximately 0.8 for inviscid flows [15]. This difference is most likely due to the differing treatment of boundary conditions between the two schemes. Despite this lower safety factor, the maximum time step remains significantly larger than the time step allowed by the fully explicit CFL criterion.

For viscous flow, the stability criterion is taken from Morgan and Peraire and is the result of a linearization of the Navier-Stokes equations. The viscous stability criterion for the PCICE algorithm is simply Peraire's stability criterion with the acoustic component removed [15]. The maximum time step allowed in viscous flow is given by:

$$\Delta t \leq \frac{C\Delta s^2 c_p \rho}{2k + |\vec{u}|\Delta s c_p \rho} \tag{148}$$

The typical safety factor for viscous flows is again 0.1. For slow viscous flows, this safety factor must be decreased to the range of 0.01 to 0.05. In these cases, the diffusive terms dominate. As these terms are highly nonlinear, the linearization performed in deriving the above stability criterion is too approximate [17]. This safety factor compares less favorably to the finite element PCICE scheme, which uses a safety factor of approximately 0.5 for slow, viscous flows [15].

### *3.6 Implementation of Boundary Conditions*

The implementation of boundary conditions are an important part of any numerical solution. In the field of fluid dynamics, the manner in which boundary conditions are treated can affect the stability, convergence and global accuracy of the numerical scheme [4]. In the PCICE algorithm, the application of boundary conditions is complicated by the fact that they must be applied multiple times within the time step. Within the finite volume PCICE scheme, this application is especially complicated due to the choice of a cell-centered finite volume scheme. Since no variables are defined at the boundary, the influence of the boundary

on the cell centered quantity must be determined. Determining the proper treatment of these boundary conditions represented some of the most difficult work encountered in this research. In this section, the treatment of boundary conditions in the PCICE-FVM scheme will be detailed. Three general types of boundary conditions will be examined. These include no flow boundary conditions, flow inlets and flow outlets. For the three general types of boundary conditions, the treatment of each of the fluxes and the pressure gradient term will be addressed.

The implementation of boundary conditions is a difficult subject in the case of computational fluid dynamics. The source of this difficulty is the fact that a system of equations (the Navier-Stokes Equations) are being solved as opposed to a single equation. Additionally, due to the hyperbolic nature of the equations, only certain variables can be specified at a given boundary. These variables are determined by a characteristic analysis of the equations at the boundary. The result of a characteristic analysis on the governing equations results in four eigenvalues, which correspond to the direction of the characteristics in the domain. For inviscid flow, these eigenvalues are $u_n$,$u_n$, $u_n + c$, $u_n - c$; where, $u_n$ is the velocity normal to the boundary and $c$ is the speed of sound [16]. At a boundary, these characteristics determine the number of variables which can be specified. The number of variables specified at a boundary is equal to the number of characteristics entering the solution domain through that boundary. Because all values must be defined at the boundary, the ideas of physical and numerical boundary conditions are useful. Physical boundary conditions are specified physical values and are independent of the solution within the domain. The number of physical boundary conditions at a given boundary is equal to the number of characteristics entering the domain. The variables not physically specified must be somehow extrapolated from the interior of the domain. These extrapolated variables correspond to numerical boundary conditions [4]. Using these concepts, the specifics of each type of boundary condition can now be explained.

The extrapolation used in this work is a simple first order extrapolation to the boundary. This extrapolation uses the fact that within the code, the cells at the boundary are roughly structured quadrilaterals. For a cell along the boundary, one of the faces will be denoted

as a boundary face. Using the local side numbering and knowing that a quadrilateral is the cell shape, the face opposite of the boundary face can be determined. The cell centered value of the boundary cell and the cell adjacent to the boundary cell across the face opposite to the boundary are then used to extrapolate to the boundary. This extrapolation can be represented by the following:

$$\Phi_b = \Phi_i - \frac{\Phi_{j,o} - \Phi_i}{|\vec{r}_{j,o} - \vec{r}_i|}|\vec{r}_b - \vec{r}_i| \tag{149}$$

where the subscript $j, o$ represents the value associated with the cell center adjacent to the face opposite of the boundary. For a uniform grid, this expression reduces to:

$$\Phi_b = \frac{3}{2}\Phi_i - \frac{1}{2}\Phi_{j,o} \tag{150}$$

which is the first order extrapolation formula often given for finite differencing [10].

### 3.6.1 Overview of Boundary Conditions Present in PCICE-FVM scheme

The PCICE-FVM scheme can be expressed as the following set of equations for illustration purposes:

$$A_i\vec{U}_i^c = A_i\vec{U}_i^n - \frac{\Delta t}{2}\sum_k[\vec{F}_{c,k}^n + \vec{F}_{c,k}^p]\cdot\vec{dl}_k + \Delta t\sum_k \vec{F}_{d,k}\cdot\vec{dl}_k + \Delta t A_i\vec{S}_i \tag{151}$$

$$\begin{aligned}\frac{A_i\delta P_i}{(\gamma-1)e_i^c} - \frac{\Delta t^2}{4}\sum_k \nabla(\delta P)\cdot\vec{dl}_k &= A_i\frac{\rho_i^c}{e_i^c}(e_i^c - e_i^n) \\ &- \frac{\Delta t}{2}\sum_k[(\rho u^c + \rho u^n)dy_k - (\rho v^c + \rho v^n)dx_k] + \frac{\Delta t^2}{2}\sum_k \nabla P^n\cdot\vec{dl}_k\end{aligned} \tag{152}$$

$$A_i\vec{U}_i^c = A_i\vec{U}_i^c - \frac{\Delta t}{2}\sum_k[\vec{F}_{u,k}^{n+1} - \vec{F}_{u,k}^p]\cdot\vec{dl}_k \tag{153}$$

where $\vec{F}_u$ is defined such that the target discretization is achieved. In a cell centered scheme, the boundaries of the domain occur at the faces of the cells. Therefore, only values evaluated at the cell face are required. By inspection of the above equations, one can see that the following values are required at the boundary. For the explicit advancement and pressure-correction equations, the convective flux ($\vec{F}_c$), the diffusive flux ($\vec{F}_d$), and the updated flux ($\vec{F}_u$) which includes the boundary pressure, are required.

The pressure Poisson equation requires knowledge of the normal gradient of pressure or the change in the pressure $\delta P$ along the boundary. In addition to this information, it also requires the mass flux at the boundary. It is in this requirement that these boundary conditions encounter issues. According to Martineau, a characteristic analysis at the boundary will show that the normal gradient in pressure can not be specified at the boundary. On top of this general statement, it can also be shown that specification of the mass flux and the normal gradient of the pressure will over specify a boundary [15]. To address this problem, a relationship between the normal gradient of the pressure and the mass flux was found. From the pressure correction equation of the momentum at the boundary, the following relationship can be developed.

$$(\rho\vec{u})^{n+1} = \rho\vec{u}^{c} - \frac{\Delta t}{2}\nabla(P^{n+1} + P^{n}) \tag{154}$$

$$(\rho\vec{u})^{n+1} = \rho\vec{u}^{c} - \frac{\Delta t}{2}\nabla(\delta P + 2P^{n}) \tag{155}$$

$$(\rho\vec{u}^{n+1} + \rho\vec{u}^{n}) = [\rho\vec{u}^{c} + \rho\vec{u}^{n}] - \frac{\Delta t}{2}\nabla(\delta P + 2P^{n}) \tag{156}$$

$$(\rho\vec{u})^{n+1/2} = [\rho\vec{u}^{c} + \rho\vec{u}^{n}] - \frac{\Delta t}{2}\nabla(\delta P + 2P^{n}) \tag{157}$$

where $\rho\vec{u}^{n+1/2}$ represents the mass flux evaluated at some time between $n$ and $n+1$. This value can related to a physical or numerical boundary condition. Taking the inner product of this equation with the normal of the face, the equation is transformed into the following:

$$(\rho\vec{u})^{n+1/2} \cdot \vec{dl} = [\rho\vec{u}^{c} + \rho\vec{u}^{n}] \cdot \vec{dl} - \frac{\Delta t}{2}\frac{\partial(\delta P + 2P^{n})}{\partial n}|dl| \tag{158}$$

The above equation can now be used in cases where the mass flux at the boundary is specified [15]. For clarity, the pressure Poisson equation is rewritten in a manner such that the boundary faces are separated out of the sums.

$$\frac{A_i \delta P_i}{(\gamma-1)e_i^c} - \frac{\Delta t^2}{4}\sum_{k\neq b}\nabla(\delta P)\cdot\vec{dl}_k - \frac{\Delta t^2}{4}\frac{\partial(\delta P)}{\partial n}_b|\vec{dl}_b| = A_i\frac{\rho_i^c}{e_i^c}(e_i^c - e_i^n)$$
$$- \frac{\Delta t}{2}\sum_{k\neq b}[(\rho u^c + \rho u^n)dy_k - (\rho v^c + \rho v^n)dx_k] - \Delta t(\rho\vec{u})_b^{n+1/2}\cdot\vec{dl} \qquad (159)$$
$$+ \frac{\Delta t^2}{2}\sum_k \nabla P^n\cdot\vec{dl}_k + \frac{\Delta t^2}{2}\frac{\partial(P^n)}{\partial n}_b|\vec{dl}_b|$$

In the above form, the manner in which the three boundary terms interact can be illustrated. Because the system of equations is a function of only the scalar variable $\delta P$, the system is fully constrained by the application of either Dirichlet or Neumann boundary conditions. Since the Neumann boundary conditions would require specification of the normal pressure gradient at the boundary, it is in this case that the mass flux at the boundary is specified. In cases where $P$, and consequently $\delta P$, is known at the boundary, the normal pressure gradient at the wall is approximated using the known pressure at the boundary and the mass flux term is extrapolated from the interior of the domain. The details of this process will be addressed in the subsequent sections.

### 3.6.2 No Flow Boundaries

No Flow boundaries are boundaries at which the normal component of velocity is equal to zero [16]. This type of boundary condition is applicable to solid free-slip walls and symmetry boundary conditions. A special subset of No Flow boundary conditions occur with viscous flow. These special no flow boundaries include isothermal no-slip walls and no-slip walls with specified heat flux. For no-slip walls, the normal component is still set to zero and tangential component of the velocity is equal to the wall velocity. Isothermal walls have a constant specified temperature and walls with a specified heat flux have a prescribed heat flux at the boundary. If this prescribed heat flux is zero, this becomes an adiabatic wall.

Since the normal component of the velocity is equal to zero, the convective flux through the boundary is also equal to zero. For Inviscid flow, this is the only physical boundary condition which can be imposed. Luckily, the only other quantity needed at the boundary is the pressure. This pressure is found using the extrapolation laid out in Equation 149.

For viscous flow, the temperature (or heat flux) and the tangential component of velocity can be specified at the boundary. For no-slip walls, the tangential component of velocity is set to the velocity of the wall. For free-slip viscous boundaries, the components of velocity are first extrapolated to the boundary. Using these values, the normal component is forced to zero, yielding a velocity which is purely tangential to the boundary. These additional specifications enter the equations through the diffusive flux at the boundary. At the boundary the following values are needed: $\tau_{xx}$, $\tau_{xy}$, $\tau_{yy}$ and $\vec{q} \cdot \vec{n}$. For no-slip viscous boundaries, the normal viscous stress is set equal to zero. The viscous shear stress at the boundary is then taken to be the shear stress evaluated at the cell center.

$$\tau_{xy,b} = \tau_{xy,i} = \mu(\frac{\partial u}{\partial y}_i + \frac{\partial v}{\partial x}_i) \tag{160}$$

$$\tau_{xx,b} = \tau_{yy,b} = 0 \tag{161}$$

On the surface this may appear to be a poor approximation, but upon more careful inspection the merit is found. In the evaluation of the cell centered viscous stress, the cell centered partial derivatives of velocity are used. These partial derivatives depend upon the velocities located at the boundary of the cells. It is in this evaluation that the proper boundary conditions on velocity are imposed (velocity set to zero). For example, on a structured mesh with a boundary at $y = 0$ and $v = 0$, this approximation for a no-slip boundary reduces to:

$$\tau_{xy,b} = \mu(\frac{u_{bo} - u_w}{\Delta y}) \tag{162}$$

where $u_{bo}$ is the velocity at the face opposite of the boundary, $u_w$ is the velocity of the wall (commonly zero), and $\Delta y$ is the height of the boundary cell. Similarly for free-slip viscous boundaries,

$$\tau_{xy,b} = 0 \tag{163}$$

$$\tau_{xx,b} = \tau_{xx,i} \qquad\qquad \tau_{yy,b} = \tau_{yy,i} \tag{164}$$

where the cell-centered viscous stresses are evaluated using the previously set velocities at the boundaries [7].

In addition to the viscous stress at the boundary, the heat flux must be found. In the case of specified heat flux, this value is given and is part of the problem statement. For isothermal boundaries, the heat flux must be approximated using the wall temperature. This is done by using a one sided derivative to evaluate the temperature gradient. Hence,:

$$\vec{q} \cdot \vec{n}_b = k_b \nabla T \cdot \vec{n}_b \tag{165}$$

$$\vec{q} \cdot \vec{n}_b \approx k_i \frac{T_b - T_i}{|\vec{r}_b - \vec{r}_i|} \tag{166}$$

With all of the flux terms determined at the boundary, the treatment of no flow boundaries in the pressure Poisson equation may be addressed. For a no flow boundary, the mass flux (momentum) at the boundary is specified to be zero. By specifying this value, the pressure at the boundary and the pressure gradient may not be specified. Therefore, in the pressure Poisson equation, these boundary terms are omitted. This omission leaves the mass flux at the boundary as the only boundary value and effectively prescribes a Neumann boundary condition for the equation. For the no flow boundary, it is set to zero. Therefore, for a cell with a face on a no flow boundary, the pressure Poisson equation is given by.

$$\begin{aligned} \frac{A_i \delta P_i}{(\gamma - 1) e_i^c} - \frac{\Delta t^2}{4} \sum_{k \neq b} \nabla(\delta P) \cdot \vec{dl}_k &= A_i \frac{\rho_i^c}{e_i^c} (e_i^c - e_i^n) \\ - \frac{\Delta t}{2} \sum_{k \neq b} [(\rho u^c + \rho u^n) dy_k - (\rho v^c + \rho v^n) dx_k] &+ \frac{\Delta t^2}{2} \sum_{k \neq b} \nabla P^n \cdot \vec{dl}_k \end{aligned} \tag{167}$$

### 3.6.3 Inlet Boundary Conditions

At an inlet boundary, the normal component of the velocity is into the domain. Inlet boundary conditions include specified mass flow inlets, specified stagnation pressure and temperature, specified static pressure and temperature, and specified Mach number. The variables that can be specified at an inlet depend on the nature of the flow at the inlet. For the subsonic case, three characteristics are entering the domain while one is exiting the

domain. Therefore, only three variables may be specified at the inlet, and the other must be extrapolated [16]. For this work, stagnation pressure and temperature inlets as well as prescribed mass flow inlets have been implemented. The calculation of boundary values for these boundary conditions will now be explained.

#### *3.6.3.1 Stagnation Pressure and Temperature Inlet*

At a stagnation inlet, the stagnation pressure and temperature are fixed. These quantities are defined as:

$$T_o = T + \frac{\vec{u} \cdot \vec{u}}{2c_p} \tag{168}$$

$$P_o = P(\frac{T}{T_o})^{-\frac{\gamma}{\gamma-1}} \tag{169}$$

where $T_o$ and $P_o$ denote the stagnation temperature and pressure. Assuming the flow is subsonic, three variables may be specified at the boundary, and the fourth must be extrapolated. In addition to the pressure and temperature, the direction of the flow is also specified [16]. To calculate the conversed and primitive variables at the boundary, the following procedure is used. First, the momentum and density are extrapolated to the boundary using Equation 149. The magnitude of the boundary momentum is then calculated. The individual components of the momentum are calculated in accordance with the specified direction at the inlet. The boundary velocity is computed by dividing the boundary momentum by the extrapolated density.

The static pressure and temperature are calculated using the definitions for the stagnation quantities. Using the static pressure and temperature, the density at the boundary is recomputed according to the equation of state:

$$\rho_b = \frac{P_b}{RT_b} \tag{170}$$

Using this density, the velocity is recomputed. Finally, the energy at the boundary is computed according to the following equation.

$$\rho e_{t,b} = \rho_b c_p T_o - P_b \tag{171}$$

The above relations are used to determine the convective fluxes at the boundary. For viscous flow, the viscous stress and heat flux must be determined. These quantities are determined in a similar manner as the no flow boundary. The heat flux is determined by using a one sided approximation of the temperature gradient. For the viscous stress, the viscous shear stress is set to zero and the cell centered normal viscous stress is set equal to the boundary normal viscous stress. This approximation is outlined below:

$$\tau_{xx,b} = \tau_{xx,i} \qquad \tau_{yy,b} = \tau_{yy,i} \tag{172}$$

$$\tau_{xx,i} = \frac{2}{3}\mu_i(2\frac{\partial u}{\partial x}_i - \frac{\partial v}{\partial y}_i) \qquad \tau_{yy,i} = \frac{2}{3}\mu_i(2\frac{\partial v}{\partial y}_i - \frac{\partial u}{\partial x}_i) \tag{173}$$

$$\tau_{xy,b} = 0 \tag{174}$$

As was the case for the no flow boundary, the partial derivatives in this equation are evaluated using the boundary velocities set from the previous relations.

The treatment of this type of inlet boundary condition in the pressure Poisson equation is relatively simple. The pressure at the inlet is treated as a known quantity. This treatment assumes that the velocities used to calculate the static pressure are correct. In practice, this assumption does not significantly effect the numerical results. By treating the pressure at the inlet as known, the boundary values of pressure and implicit change in pressure, $\delta P$ are known values. The pressure gradients at the boundary are then approximated using the following equations.

$$\frac{\partial(P^n)}{\partial n}_b = \frac{P_b - P_i}{|\vec{r}_b - \vec{r}_i|} \tag{175}$$

$$\frac{\partial(\delta P^n)}{\partial n}_b = \frac{\delta P_b - \delta P_i}{|\vec{r}_b - \vec{r}_i|} \tag{176}$$

Since the pressure is specified at the boundary, the mass flux at the boundary can not be specified. Instead, the boundary value of the mass flux is extrapolated, just as it is

for the convective flux. The pressure Poisson equation for an inlet boundary using the approximation outlined above is given below:

$$\begin{aligned}\frac{A_i \delta P_i}{(\gamma-1)e_i^c} - \frac{\Delta t^2}{4}\sum_{k\neq b}\nabla(\delta P)\cdot\vec{dl}_k + \frac{\Delta t^2}{4}\frac{\delta P_i}{|\vec{r}_b-\vec{r}_i|}|\vec{dl}_b| = A_i\frac{\rho_i^c}{e_i^c}(e_i^c - e_i^n) \\ - \frac{\Delta t}{2}\sum_{k\neq b}[(\rho u^c + \rho u^n)dy_k - (\rho v^c + \rho v^n)dx_k] - \Delta t(\rho\vec{u})_{ext.}^{n+1/2}\cdot\vec{dl} \\ + \frac{\Delta t^2}{2}\sum_k \nabla P^n\cdot\vec{dl}_k + \frac{\Delta t^2}{2}\frac{P_b - P_i^n}{|\vec{r}_b-\vec{r}_i|}|dl_b| + \frac{\Delta t^2}{4}\frac{\delta P_b}{|\vec{r}_b-\vec{r}_i|}|\vec{dl}_b|\end{aligned} \tag{177}$$

where $\delta P_b$ is zero for a steady specified pressure and $\rho\vec{u}_{ext.}^{n+1/2}$ is given by:

$$\rho\vec{u}_{ext.}^{n+1/2} = \frac{\rho\vec{u}_{ext.}^{c} + \rho\vec{u}_{ext.}^{n}}{2} \tag{178}$$

*3.6.3.2 Mass Flow Inlet*

For specified mass flow inlets, the density and velocity at the boundary are given. The two components of velocity and the density represent the three physical boundary conditions enforced at the inlet. The rest of the variables are otherwise determined from the solution in the interior of the domain. To determine the other variables at the boundary, the pressure is extrapolated from the interior of the domain. Using this extrapolated pressure and the density, the temperature at the boundary is calculated. This temperature is then used to calculate the total energy of the boundary. The treatment of the diffusive fluxes at the inlet is identical to the treatment used for the stagnation pressure inlet.

The main difference between mass flow inlets and stagnation inlets is the treatment of the boundary conditions in the pressure Poisson equation. In the case of the mass flow inlet, the pressure at the boundary is not defined. Therefore, a Neumann boundary condition must be applied at the boundary. As is the case with the no flow boundary condition, the Neumann boundary condition for pressure is replaced by the mass flow at the boundary, which is a known and constant value. As such, the pressure Poisson equation for a cell on the mass flow inlet is given by the following:

$$\frac{A_i \delta P_i}{(\gamma-1)e_i^c} - \frac{\Delta t^2}{4}\sum_{k\neq b}\nabla(\delta P)\cdot\vec{dl}_k = A_i\frac{\rho_i^c}{e_i^c}(e_i^c - e_i^n)$$
$$- \frac{\Delta t}{2}\sum_{k\neq b}[(\rho u^c + \rho u^n)dy_k - (\rho v^c + \rho v^n)dx_k] - \Delta t(\rho\vec{u})_b + \frac{\Delta t^2}{2}\sum_k \nabla P^n\cdot\vec{dl}_k \tag{179}$$

where $(\rho\vec{u})_b$ is the specified mass flux at the boundary.

### 3.6.4 Outlet Boundary Conditions

Outlet boundary conditions are essentially the opposite of inlet boundary conditions. At an outlet, the normal component of the velocity is out of the domain. For a subsonic outlet, this definition means that three characteristics are out of the domain and one characteristic is into the domain. For a supersonic outlet, all of the characteristics exit the domain. For the subsonic case, the static pressure at the outlet is typically specified [16]. The details of each outlet type will be given in the following sections. In the numerical implementation of this boundary, the flow conditions at the outlet must be determined and the proper boundary condition applied.

#### *3.6.4.1 Subsonic Outlet with Specified Static Pressure*

This type of outlet is similar to the stagnation inlet outlined before. For this boundary, three characteristics exit the domain and one enters. Therefore, only one variable may be specified at the outlet. In this case, the static pressure is specified. To determine the other boundary variables, extrapolation is used [16]. At the exit, the momentum and density are first extrapolated to the boundary. From these values, the velocity is calculated. The extrapolated density and specified pressure are then used to calculate the temperature at the outlet from the equation of state. The total energy at the boundary is then calculated using the following formula:

$$\rho e_{t,b} = \rho_b(c_v T_b + \frac{\vec{u}_b\cdot\vec{u}_b}{2}) \tag{180}$$

These boundary variables are then used to calculate the convective flux at the boundary. The treatment of diffusive fluxes at the outlet is identical to the treatment used at inlets

and will not be outlined again. Finally, the pressure Poisson equation for a cell on the outlet takes the same form as the equation for the stagnation inlet. The normal pressure gradient at the boundary is approximated using the cell centered pressure and the specified boundary pressure. The mass flux at the boundary is then simply extrapolated. The pressure Poisson equation for a cell at a subsonic outlet is identical to the equation used for the stagnation inlet boundary.

#### *3.6.4.2 Supersonic Outlet*

For an outlet where the fluid velocity is greater than the local speed of sound, all four characteristics exit the domain. In this case, no variables may be specified at the boundary [16]. For this work, the momentum, density, and pressure are extrapolated to the boundary. The momentum and density are then used to calculate the boundary velocity. The boundary temperature and energy are then calculated according to the equation of state and Equation 180. The convective fluxes and diffusive fluxes are calculated identically to the subsonic outlet.

The pressure Poisson equation for a supersonic outlet is slightly different than the subsonic case. For the supersonic outlet, the pressure at the boundary is not specified. As such, the normal gradient of the pressure at the boundary is omitted from the equation. Additionally, since the mass flux at the boundary is not known, the extrapolated mass flux is used. The pressure Poisson equation for a cell at a supersonic outlet is therefore given by:

$$\begin{aligned}\frac{A_i \delta P_i}{(\gamma-1)e_i^c} - \frac{\Delta t^2}{4}\sum_{k\neq b}\nabla(\delta P)\cdot\vec{dl}_k &= A_i\frac{\rho_i^c}{e_i^c}(e_i^c - e_i^n) \\ &- \frac{\Delta t}{2}\sum_{k\neq b}[(\rho u^c + \rho u^n)dy_k - (\rho v^c + \rho v^n)dx_k] - \Delta t(\rho\vec{u})_{ext.}^{n+1/2}\cdot\vec{dl} \\ &+ \frac{\Delta t^2}{2}\sum_k \nabla P^n\cdot\vec{dl}_k\end{aligned} \tag{181}$$

# CHAPTER IV

# CONJUGATE HEAT TRANSFER

The term conjugate heat transfer (CHT) is given to problems in which the heat transfer between a solid phase and fluid phase is not known a priori, but has a significant effect on the system. Conjugate heat transfer problems include the analysis of heat exchangers, aerodynamic heating, and problems involving the cooling or heating of solids [26]. Solving these problems numerically is a relatively new subject of research. In this chapter, the previous advances in solving these types of problems will be outlined. Following this outline, the type conjugate heat transfer implemented in this work will be explained and its inclusion in the finite volume formulation will be detailed.

## *4.1 Background*

The first attempts at conjugate heat transfer simply involved the coupling of a fluid dynamics solver with a solver used for solid state conduction. Initially, this coupling was done through boundary conditions on the energy equation. Initially, heat transfer coefficients were calculated and the heat flux at the boundary was adjusted accordingly. As fluid dynamic solvers became more advanced and boundary layers began to be resolved, this heat flux could be calculated using Fourier's law and the surface temperature of the solid. In Reference [23], the authors describe a method for performing transient conjugate heat transfer calculations which accurately represents this approach. In their method, at the beginning of the time step, the fluid variables are advanced using the surface temperature at the previous time step. Using these updated fluid variables, the heat flux entering the solid is calculated. This new heat flux is used to update the solid temperature and surface temperature. Finally, the fluid variables are corrected using the updated solid temperature and code advances to the next time step. This loose coupling matched experimental data reasonably well; however, for a given time step, the heat flux entering the solid is not necessarily equal to the heat flux leaving the fluid. This temporal coupling was further

investigated by Webster. In this work, the solid temperature was found to always lag the fluid values, causing the boundary condition to be fully explicit in time [27].

An improvement can be made to this coupling in the case of steady state problems by iterating between the fluid and solid until both the heat flux and tempeature are matched at the boundary. Additionally, by setting the velocity within the solid as zero, the domains can be combined into one frame work. This is the approach used by the authors of References [6] and [8]. In these works, the fluid and solid phase are treated as part of the same domain by setting the thermodynamic and solution variables of the solid appropriately. Although the fluid and solid are combined in the same domain, the coupling between phases is treated with boundary conditions. As the system iterates to achieve steady state, special attention is given to the interface to ensure that as the system approaches steady state, temperature and heat flux are continuous across the boundary [8]. A similar iterative process could be applied to transient problems; however, it would be required at each time step and thus would be computationally expensive.

Recently, a more rigorous treatment of conjugate heat transfer has been developed by Martineau at the Idaho National Lab. This form of CHT represents a strong, mathematical coupling between the physics of fluid flow with solid phase heat conduction. This approach eliminates the need for energy boundary conditions between the fluid and solid phase. Eliminating this need allows for the temperature and heat flux at the boundary to be continuous for each time these terms are evaluated [18]. The key observation in this approach is the following. When the velocity is equal to zero, the governing equations reduce to:

$$\frac{\partial \rho}{\partial t} = 0 \tag{182}$$

$$\frac{\partial \rho \vec{u}}{\partial t} = 0 \tag{183}$$

$$\rho c_v \frac{\partial T}{\partial t} = \nabla (k \nabla T) + i \tag{184}$$

where Fourier's law has been used for the conductive heat flux, and the internal energy has been replaced by the product of specific heat and temperature. Cast in this form, the energy equation is identical to the heat equation in a solid medium. The conclusion which

can be drawn from this observation is that the energy equation is equally valid throughout the entire domain, whether fluid or solid. Consequently, a single discretization can be applied in the fluid phase, the solid phase and at the interface between phases [18].

In order to apply this approach in a numerical formulation, two requirements are needed. First, the condition that the velocity is zero within the solid must be strictly enforced. The second is that the discretization used for the conductive heat flux must allow for large variations in the thermal conductivity throughout the domain. It should be noted that this requirement applies strictly to the spatial discretization, allowing for this treatment of CHT to be applied in a number of temporal discretizations. Allowing for dramatic changes in thermal conductivity is not a trivial requirement of a spatial discretization. A dramatic change in thermal conductivity across a boundary implies the presence of a large gradient in temperature.

## *4.2 Implementation of CHT into PCICE Finite Volume Scheme*

### 4.2.1 Approximation of Heat Flux

The implementation of conjugate heat transfer into the finite volume formulation outlined in the previous section was done through the approximation of the heat flux vector. This term requires the approximation of the temperature gradient. Typically, the temperature gradient across a face would be assumed to be continuous. In regions where the thermal conductivity does not change significantly, this approximation would be appropriate. In the context of the previous outlined work, this approximation would be given by [7]:

$$\vec{q}_k \cdot \vec{dl}_k = -k_k \frac{\partial T}{\partial n}_k |\vec{dl}_k| \tag{185}$$

$$\vec{q}_k \cdot \vec{dl}_k = -k_k \frac{T_j - T_i}{|\vec{r_j} - \vec{r_i}|} |\vec{dl}_k| \tag{186}$$

where $k_k$ is the thermal conductivity interpolated to the face. This approximation assumes that the temperature distribution between cell center $j$ and $i$ is linear. In regions where the thermal conductivity changes rapidly, such as the interface between fluid and solid, this assumption is no longer valid. To derive an appropriate approximation, the physical conditions satisfied at the face between cells must be reconsidered. To place this

in a more familiar context, the interface between two cells of differing materials will be considered. At the interface between materials, the temperature across the interface must be continuous and the heat flux exiting one material must be equal to the heat flux entering the adjacent material. These relations are given below:

$$q''_{b^-} = -q''_{b^+} \tag{187}$$

$$T_{b^-} = T_{b^+} \tag{188}$$

where $\vec{q_b}$ and $T_b$ are the heat flux and temperature at the interface, respectively. Utilizing Fourier's law, the heat flux in the above equation can be rewritten as follows.

$$q''_{b^-} = -k_- \frac{\partial T}{\partial n}|_{b^-} \qquad\qquad q''_{b^+} = -k_+ \frac{\partial T}{\partial n}|_{b^+} \tag{189}$$

where $k_-$ and $k_+$ denote the thermal conductivity on each side of the interface. The normal temperature gradient in the above equation can now be approximated in a similar manner as Equation 186.

$$q''_{b^-} = -k_- \frac{T_b - T_i}{|\vec{r}_b - \vec{r}_i|} \tag{190}$$

$$q''_{b^+} = -k_+ \frac{T_b - T_j}{|\vec{r}_b - \vec{r}_j|} \tag{191}$$

In these equations, the temperature gradient has been approximated using the temperature at the boundary and the cell centered temperature. The requirement that the temperature at the boundary be continuous has been implicitly assumed in these approximations. Equation 187 can now be written in the following form.

$$k_- \frac{T_b - T_i}{\Delta s_{i \to b}} = -k_+ \frac{T_b - T_j}{\Delta s_{j \to b}} \tag{192}$$

where $\Delta s$ is the distance from the cell center to the boundary. By taking the thermal conductivity to be constant over a cell ($k_- = k_i$), the temperature at the boundary can be determined using only cell centered values. The temperature at the boundary is given by:

$$T_b = \frac{k_j T_j \Delta s_{i\to b} + k_i T_i \Delta s_{j\to b}}{k_j \Delta s_{i\to b} + k_i \Delta s_{j\to b}} \tag{193}$$

This boundary temperature can then be used in Equations 190 and 191 to determine the heat flux passing through the interface. Although this approximation was derived by considering the interface between two different materials, it is equally valid for all cells within the domain. The heat flux through any face in the domain is therefore approximated by:

$$\vec{q} \cdot \vec{dl}_k = -k_i \frac{T_b - T_i}{|\vec{r}_b - \vec{r}_i|} |\vec{dl}_k| \tag{194}$$

This approximation is used within the explicit diffusion step of the PCICE-FVM scheme to approximate the heat conduction term.

### 4.2.2 Numerical Treatment

In order to simulate problems with fluid and solid phases, several precautions must be taken. Within the frame of the fluids solver implementing this scheme, each cell is assigned a material flag. For cells in the solid body, the velocity is ensured to be zero. In order to achieve this, the velocity within the solid is initialized to be zero. The convective fluxes within the solid are then never calculated within the running of the code. With these fluxes as zero, the velocity within the solid never becomes nonzero.

The next issues that must be addressed are the treatment of pressure within the solid and the treatment of boundary conditions in the combined system. Since the pressure at a fluid/solid interface is required for the solution to the momentum equation within the fluid, these two issues are tightly linked. The thermodynamic pressure within the solid is irrelevant to the types of problems being analyzed with this formulation. Therefore, the pressure Poisson equation within the solid is not valid in its present form. To avoid this problem, the change in pressure within the solid is forced to be zero. Hence, the pressure within the solid remains the pressure with which it was initialized. The problem with this technique is that at the interface between a solid and fluid, the pressure interpolated at the boundary will yield a nonphysical result. This leads to the issue of how to treat boundary conditions within the combined systems.

At fluid/solid interfaces, the convective and diffusive fluxes must be evaluated, just as in the rest of the domain. In terms of the fluid domain, these interfaces represent places where the convective and diffusive fluxes must satisfy certain criteria. The convective fluxes through the face must be zero and the viscous fluxes must correspond to those of a no-slip boundary. At the same time, the flux corresponding to heat conduction can be treated uniformally throughout the domain and no special attention is given to the interface. To reconcile these competing factors, a compromise was reached. For fluid cells located at the interface, boundary conditions for the convective and viscous fluxes are set according to the definitions in Section 3.6. For the solid cells, the only variable allowed to vary is the internal energy and the only flux term present is due to conduction. Since the conductive flux requires no special treatment, no boundary conditions are enforced on this flux. Through this compromise, the physical conditions needed for the fluid at the interface are directly enforced; however, the heat transfer between the solid and fluid is treated without a boundary condition. It should be noted that this compromise is only necessary for a cell-centered scheme. The value of the convective flux and viscous flux at the interface will take on the appropriate value if the velocity along the interface is zero. Since no variables are stored at the interface, this condition can not be guaranteed and boundary conditions for the flow must be enforced. For finite element methods or cell-vertex methods, the variables along the interface can be designated as belonging to the solid. As such, the velocity is ensured to be zero. With proper treatment of the pressure Poisson equation, the interface can be treated completely without boundary conditions. This is the case in the finite element implementation of this conjugate heat transfer formulation [18].

Finally, since the heat conduction terms are treated explicitly, the stability of the numerical scheme within the solid should be examined. In the solid, the maximum time step requirement outlined in Equation 148 becomes:

$$\Delta t \leq \frac{C \Delta s^2 c_p \rho}{2k} \tag{195}$$

This stability is identical to the stability criteria used in forward time, center space formulations used to solve the heat equation [11], given by:

$$d = \frac{k\Delta t}{\rho c_v \Delta s^2} \leq \frac{1}{2} \tag{196}$$

It should be noted that within a solid, $c_p$ is equal to $c_v$. Therefore, the criteria used to determine the maximum time step for viscous flow is valid within the solid. In practice, the time step is typically limited by the maximum time step allowed in the solid, giving some motivation for treating the conduction terms in an implicit manner.

# CHAPTER V

# RESULTS

Using the finite volume PCICE scheme developed in the previous section, a fortran program was written to implement the scheme and to confirm its validity. Each of these problems were selected to highlight and confirm certain aspects of the formulation. To confirm the accuracy of the Cartesian formulation and analyze the temporal performance of the scheme, the Riemann Shock Tube was solved. In order to test the axisymmetric formulation and the implementation of boundary conditions, a converging-diverging nozzle was analyzed over a variety of flow conditions and each test case was compared against an analytical solution. In order to test the formulation for solid phase heat transfer, a conduction benchmark problem inspired by the Reed Problem in radiation transport was constructed. This problem consisted of a series of material slabs with alternating high and low thermal conductivity and thermal capacitance. Finally, to test the coupling of solid conduction to fluid flow, cross flow over fuel material was examined and validated.

## *5.1 Shock Tube*

The first test problem analyzed is the classic Riemann Shock Tube. In this problem a large discontinuity in pressure is initialized. The solution to this problem can be found by analyzing the Euler equations (inviscid flow equations). The numerical solution to this problem was found using the Cartesian formulation outlined in the previous section. Since the classic problem is inviscid, the diffusive fluxes were left out of the equation. Since this problem is completely one dimensional, the analytical solution to the problem is derived through analysis of jump conditions for shock and rarefaction waves. The shock tube was analyzed for three main reasons. First, the numerical solution may be compared to the exact solution, proving that the finite volume formulation properly solves the Euler equations. Second, the performance of Jameson's artificial viscosity in capturing shocks can be illustrated. Finally, since the problem is transient in nature, the temporal behavior

of the scheme can be characterized. Toward this end, a temporal convergence study was performed.

### 5.1.1 Analytical Solution

The solution to the Riemann Shock tube is a classic problem in the field of gas dynamics and the analytical solution is derived in numerous texts such as Reference [10]. A full derivation of the solution will not be given in this paper. Instead, a qualitative overview of the solution will be outlined. There are many variants of the Riemann Shock tube. The specific problem analyzed is the following. The tube itself is 100 m long and 1 m high with an axis aligned with the x coordinate direction. Both ends of the tube are sealed. Initially, a diaphragm is located at $x = 40m$. To the left of the diaphragm, air is contained at 10 atmospheres (1013250 Pa) and to the right, the pressure is initially 1 atmosphere (101325 Pa). The temperature is initially 300 K throughout the tube. At time $t = 0$, the diaphragm bursts, causing a shock wave to propagate into the low pressure side of the gas. The speed at which this shock propagates through the low pressure fluid is equal to the sound speed plus the material velocity behind the shock. At the same time as this shock wave propagates through the low pressure fluid, a rarefaction wave is propagated into the high pressure region. As this wave passes over material, the pressure is lowered. This rarefaction wave is propagated at a speed equal to the material velocity behind the wave minus the local sound speed behind the wave. The last characteristic of the shock tube occurs in the density. In a plot of the density along the length of the tube, two discontinuities are observed. The right most discontinuity corresponds to the shock wave. The second occurs where a discontinuity in temperature is observed. This discontinuity is known as a contact discontinuity and has the following cause.

For this specific problem, the temperature is initially uniform. As the pressure in the right side of the tube is increased, the temperature also increases. The opposite is true of the left side. Therefore, a discontinuity in temperature initially develops at the location of the diaphragm. As the solution evolves, this discontinuity in temperature is simply convected along with the material. This discontinuity in temperature then causes a discontinuity in

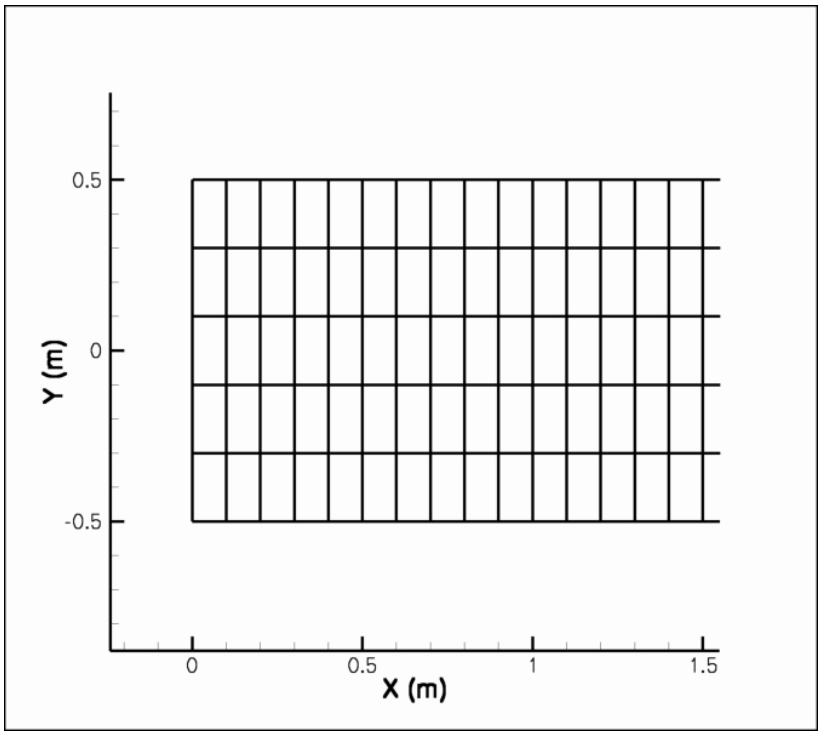


**Figure 1:** Representative Mesh for Shock Tube

density due to the equation of state. This discontinuity moves with the material velocity [10].

### 5.1.2 Numerical Solution

The numerical solution to this problem was found using the Cartesian finite volume formulation detailed in the previous sections. First, the appropriate geometry was meshed using a series of quadrilaterals. Five quadrilaterals were placed along the width of the tube and one thousand cells were placed along the length. A representative part of this mesh can be found in Figure 1.

For this simulation, the problem was carried out to a time of $t = 0.08s$. The CFL number was equal to 0.1 and the values for the Jameson coefficients are 1.8 for the second order term and 0.02 for the fourth order term. Since the problem is adequately described by inviscid flow, the simulation was performed without evaluating the diffusive fluxes. Additionally, free-slip, adiabatic wall boundary conditions were applied on all four sides of the domain. The two dimensional solution for pressure is given in Figure 2. As the figure shows, the pressure, and indeed all the solution variables, are uniform along the width of the shock

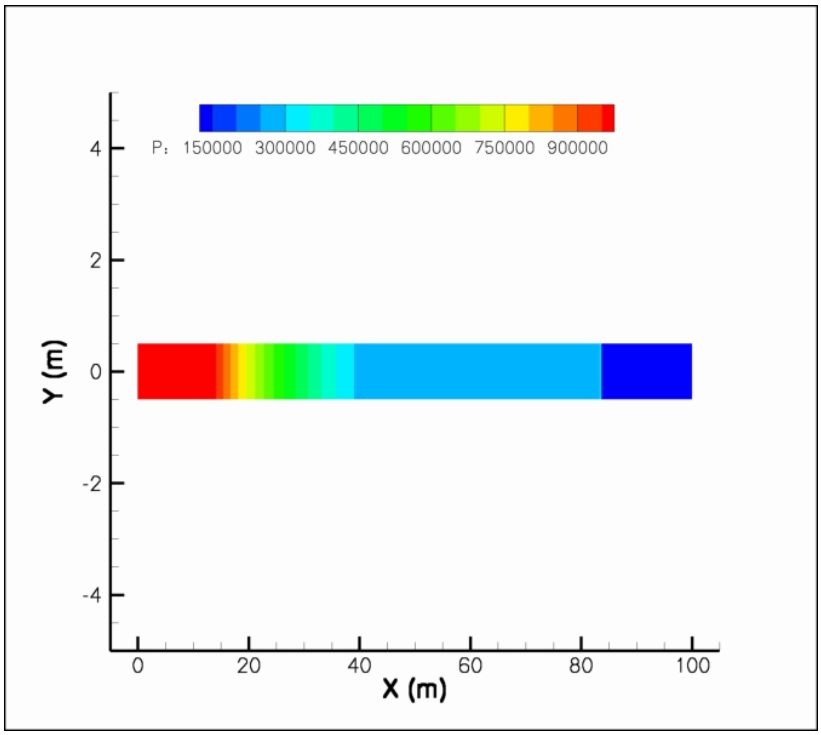


**Figure 2:** Pressure Counter at t=0.08 s

tube. Hence, the numerical solution correctly preserves the one dimensional nature of the problem. The characteristics of the solution can be better seen with a one dimensional representation of the solution variables along the centerline of the shock tube.

In addition to clarity, the one dimensional representation can be shown directly with the analytical solution to the problem. Figure 3 shows the centerline density plotted with the analytical solution for density.

As the above figure shows, the agreement with the analytical solution is quite good. The deviation from the analytical solution occurs at the main shock and the contact discontinuity. At the shock, an overshoot in the numerical solution is observed. The shock itself is sharp and occurs over roughly 8 computational cells. At the contact discontinuity, the discontinuity appears smeared over many cells. This effect can be attributed to the artificial viscosity used within the formulation. Since this discontinuity is the first to appear, the dissipation applied to it is based on the initial pressure discontinuity and is therefore large. As the solution evolves, this initial dissipation is never corrected and appears in the final solution. Figures 4 through 6 show the pressure, velocity and temperature plotted along with the analytical solution for these variables. As Figures 4 and 5 show, the overshoots

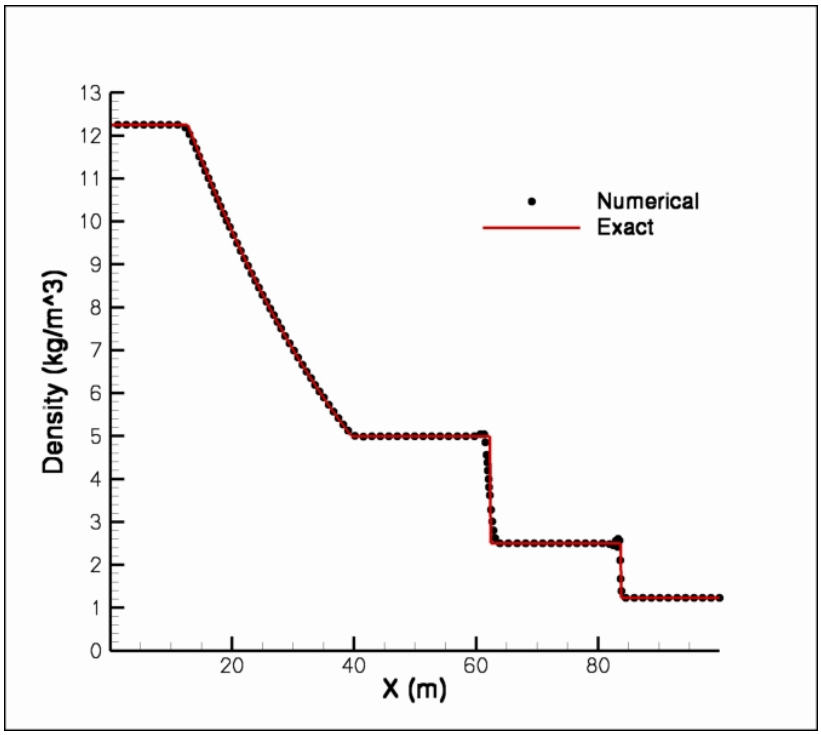


**Figure 3:** Centerline density with Analytical solution at t=0.08 s

in the pressure and velocity shocks are more pronounced than the overshoot seen in density. These relatively large overshoots cause the large overshoot in the temperature shock. Again, the shocks themselves are sharp and occur over only a few cells. In Figure 6, the discontinuity in temperature corresponding to the contact discontinuity in density appears smeared out over a number of cells. Also, an undershoot is seen at the left side of this discontinuity. This undershoot leads to a small overshoot in density at the left side of the contact discontinuity.

### 5.1.3 Temporal Convergence Study

In order to test the temporal behavior of the PCICE algorithm, a temporal convergence study was performed using the shock tube. For this study, the time step was initially set to a constant value corresponding to a CFL number of approximately 0.1. The time step was then reduced by a factor of two and the simulation was run again. For each value of the time step, the total pressure within the shock tube was calculated for use as an error measure. The total pressure was calculated using the following isentropic flow relation.

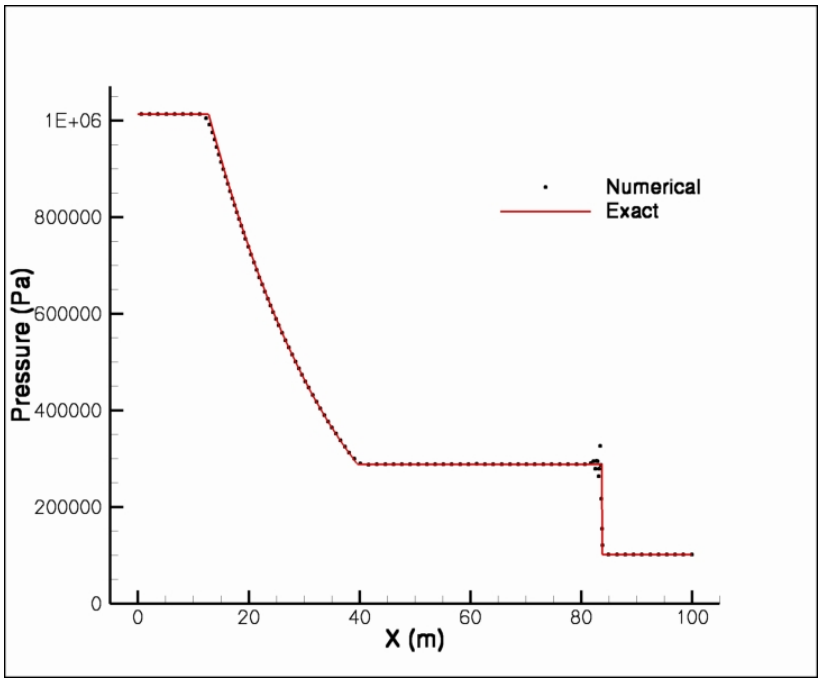


**Figure 4:** Centerline Pressure plotted with Analytical Solution at t=0.08 s

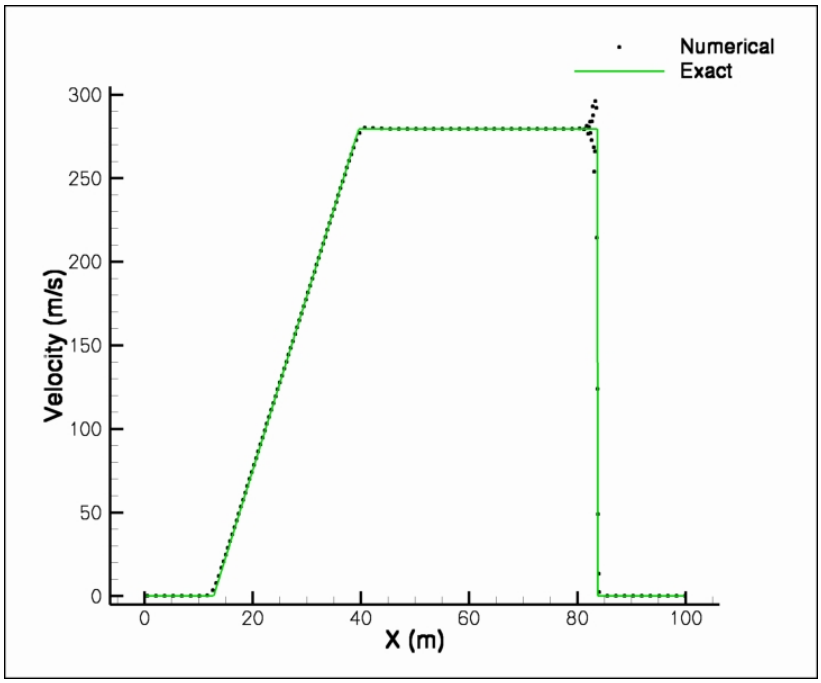


**Figure 5:** Centerline Velocity plotted with Analytical Solution at t=0.08 s

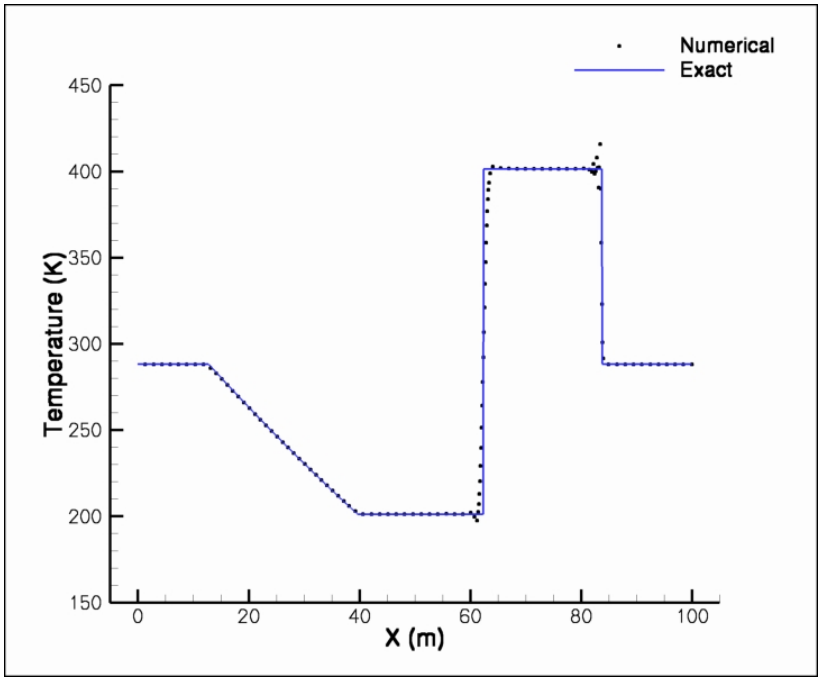


**Figure 6:** Centerline Temperature plotted with Analytical Solution at t=0.08 s

$$p_t = p(1 + \frac{\gamma - 1}{2} Ma^2)^{\frac{\gamma}{\gamma - 1}} \tag{197}$$

The integral of this value was then used as the norm to compare the results of the various values of the time step. This norm was calculated in the discrete formulation as the following:

$$f = \frac{\int_V p_t dV}{\int_V dV} \tag{198}$$

$$f \approx \frac{\sum_i p_{t,i} V_i}{\sum_i V_i} \tag{199}$$

In order to determine the temporal rate of convergence, three different values of the time step were chosen. These time steps roughly correspond to a CFL number of 0.1, 0.05 and 0.025. In order to isolate the effect of temporal truncation error, the artificial dissipation coefficient has to be set to zero. This action is required because the amount of dissipation applied to the solution is loosely dependent on the time step chosen. This varying of the dissipation caused the temporal convergence study to yield invalid results [24]. With the

**Table 1:** Results of Temporal Convergence Study

| $\Delta t$ (s) | f $(Pa)$ |
|---|---|
| $2.0 \times 10^{-5}$ | $5.054530 \times 10^5$ |
| $1.0 \times 10^{-5}$ | $5.025057 \times 10^5$ |
| $5.0 \times 10^{-6}$ | $5.018111 \times 10^5$ |
| 0.0 | $5.015968 \times 10^5$ |
| $p = 2.085013$ | |

artificial dissipation coefficients set to zero, the temporal convergence study was carried out. Using the results of this study, the temporal order of accuracy is determined by the following formula:

$$p = \frac{ln\left(\frac{f_3 - f_2}{f_2 - f_1}\right)}{ln(r)} \tag{200}$$

where $f_3$ is the norm associated with the largest time step and $r$ is the refinement ration (two for the purposes of this work) [24]. Table 5.1.3 summarizes the results of the convergence study.

The order of convergence found in the study was 2.085. This value agrees with the temporal order of convergence found with the finite element PCICE scheme. This result merely confirms that the PCICE algorithm is indeed second order accurate and the temporal accuracy is independent of spatial discretization. It should be noted that this result is valid only for inviscid calculations. The finite volume formulation presented previously treated viscous terms as first order in time. For flow in which these terms dominate, the afore mentioned order of accuracy will not be valid. Also located in Table 5.1.3 is the norm resulting from Richardson extrapolation. This value represents the norm which would be achieved if an infinite number of time steps were used to construct the transient solution. The extrapolated quantity is determined using the equation [24]:

$$f_{\Delta t=0} \approx f_1 + \frac{f_1 - f_2}{r^p - 1} \tag{201}$$

Figure 7 contains the four normalized pressure values plotted against time step. These pressure values have been normalized by the Richardson extrapolation value.

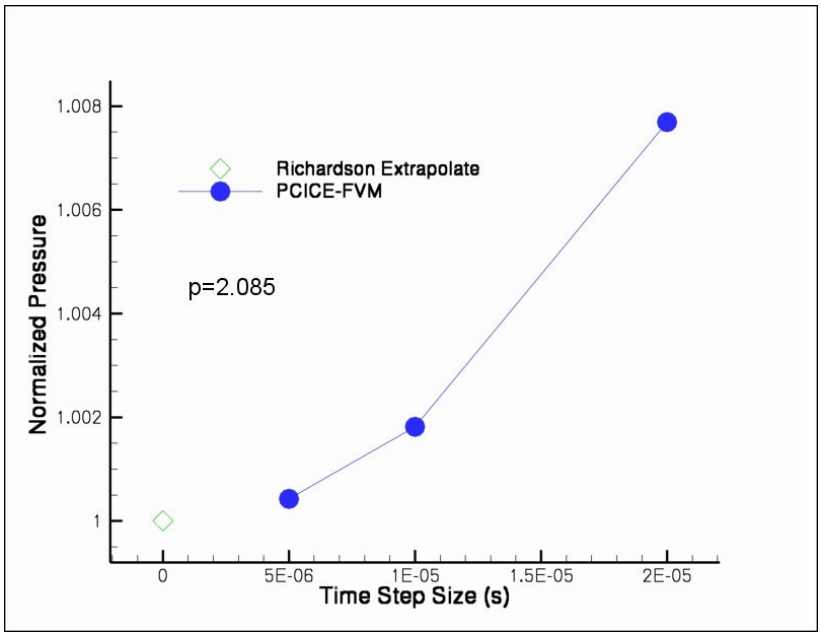


**Figure 7:** Normalized pressure vs. time step size

As the figure shows, the normalized pressures converge toward the Richardson extrapolate in a quadratic manner.

### *5.2 Converging Diverging Nozzle*

In addition to the shock tube, the flow within a converging diverging nozzle was studied. For this problem, three different cases were explored. These three cases corresponded to subsonic flow, supersonic flow and supersonic flow with a standing shock. This problem was selected for a variety of reasons. First, the performance of the axisymmetric formulation over a variety of flow regimes could be accessed. To confirm the accuracy of this formulation, the numerical solution is compared with an analytical solution derived from one dimensional, isentropic flow relations. Second, the implementation of inlet and outlet boundary conditions was examined. Finally, a spatial convergence study was performed using the supersonic case.

### 5.2.1 Problem Description

The converging-diverging nozzle used in this test problem is part of a series of bench mark problems provided by NASA. The full problem description and analysis can be found in Reference [25]. The nozzle itself is 10 inches long with an inlet cross-section of 2.5 $in^2$ and an outlet cross-section of 1.5 $in^2$. The cross-section of the nozzle is described by the following equation.

$$A = \begin{cases} 1.75 - 0.75cos[(0.2x - 1.0)\pi], & \text{if } x < 5.0; \\ 1.25 - 0.25cos[(0.2x - 1.0)\pi], & \text{otherwise.} \end{cases} \tag{202}$$

Using this formula, the geometry seen in Figure 8 was generated. At the inlet, the stagnation pressure and temperature are fixed at 1.0 psi and $100°R$, respectively. The outlet static pressure was then varied to achieve the appropriate flow conditions in the nozzle. For an exit pressure of $0.89psi$, the flow is subsonic throughout the nozzle. For an exit pressure of $0.75psi$, the flow enters the nozzle subsonically but transitions to super sonic. As the flow further expands in the diverging portion of the nozzle, a stationary shock forms. The flow then exits the nozzle subsonically. Finally, for an exit pressure of $0.16psi$, the flow enters the nozzle subsonically and exits super sonically without forming any shocks. Analytical solutions to these problems can be found using the one dimensional, isentropic flow equations. These one dimensional solutions correspond closely with the centerline solution of the two dimensional problem. The analytical solutions were taken from Reference [25].

In order to solve this problem numerically, the geometry seen in Figure 8 was meshed using 3200 cells, giving a mesh spacing of approximately 0.05 in. Sixteen cells where used along the radial direction and 200 cells were used in the axial direction. A representative portion of the grid can be seen in Figure 9

The problems were solved inviscidly and the upper and lower boundary was assigned free-slip, adiabatic wall boundary conditions. As stated previously, the symmetry boundary condition is equivalent to the free-slip, adiabatic wall. The inlet has a prescribed pressure and temperature boundary condition and the outlet is a static pressure boundary condition.

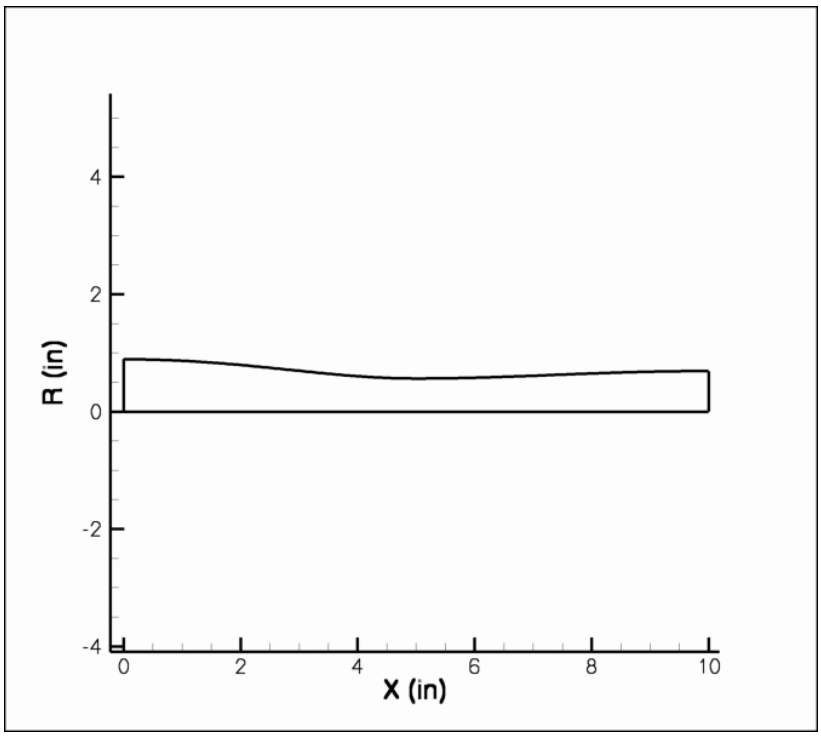


**Figure 8:** Converging-Diverging Nozzle Geometry

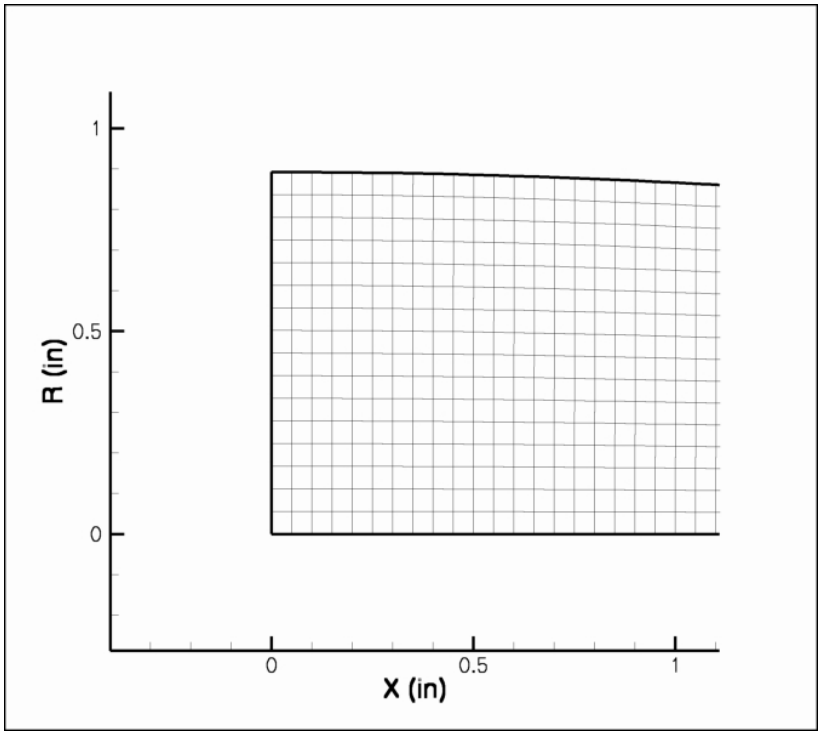


**Figure 9:** Representative Mesh for Converging-Diverging Nozzle

For the supersonic case, the outlet boundary condition is changed to a supersonic outlet when the flow becomes greater than Mach one. The steady state solutions to these problems were found by building up the solution in time until the maximum error value within the domain was less than a specified tolerance ($10^{-6}$). The error within the domain was calculated based on the relative error in the pressure. The stopping criteria for steady state problems is given below:

$$\epsilon = ||\frac{|P_i^{n+1} - P_i^n|}{P_i^{n+1}}||_{\text{inf}} \leq 10^{-6} \tag{203}$$

Additionally this solution was found using a CFL safety factor of 0.1 and smoothing coefficients of 0.5 for $\kappa_2$ and 0.008 for $\kappa_4$. The solution for each of the three flow cases is now presented, followed by the results of the spatial convergence study.

### 5.2.2 Subsonic Case

For this case, the flow is subsonic throughout the nozzle. As expected, the flow accelerates in the converging portion of the nozzle and decelerates in the diverging portion. Figure 10 shows the pressure contour for this flow case. The counter for the Mach number is shown in Figure 11.

As these plots show, the solution performs as expected. The Mach number is maximum at the throat of the nozzle and the pressure is at a minimum. Additionally, the solution is approximately constant in the radial direction. This fact allows for the solution to be represented one dimensionally. For this representation, the centerline variables were used. By representing this problem one dimensionally, the numerical solution can be compared to a one dimensional analytical solution. Figures 12 and 13 show the one dimensional representations of pressure and Mach number, both compared with the analytical solutions.

As these plots show, the numerical solution compares well with the analytical solution. The closest agreement occurs near the inlet and outlet and the maximum error is encountered in the vicinity of the throat of the nozzle. This maximum error is approximately 1.5 percent. The Mach number is under predicted at the throat, consequently causing the pressure to be over predicted. It is likely that this disagreement is due to the artificial

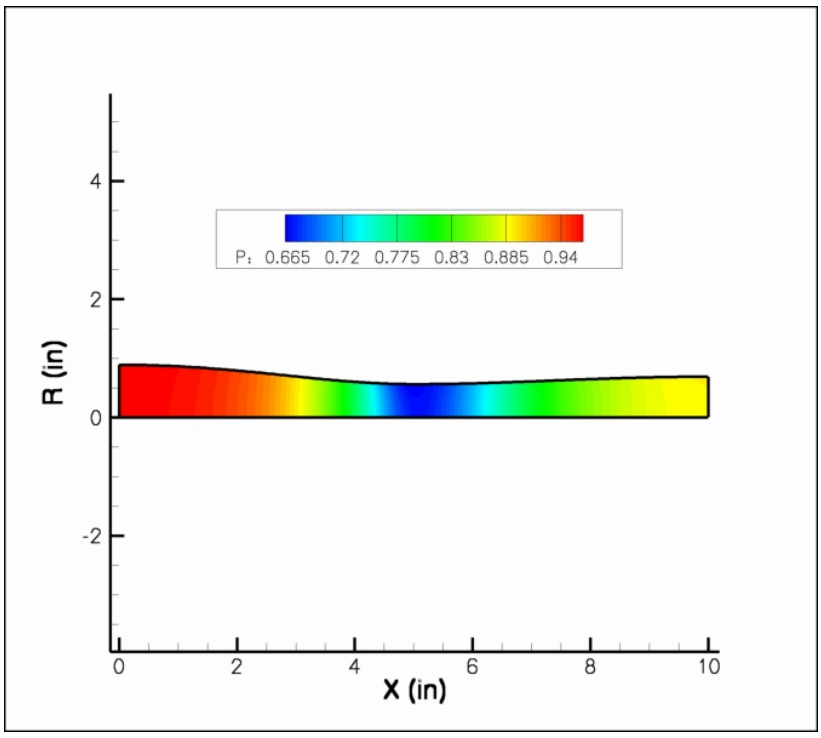


**Figure 10:** Steady State Pressure Contour Plot for Subsonic Case

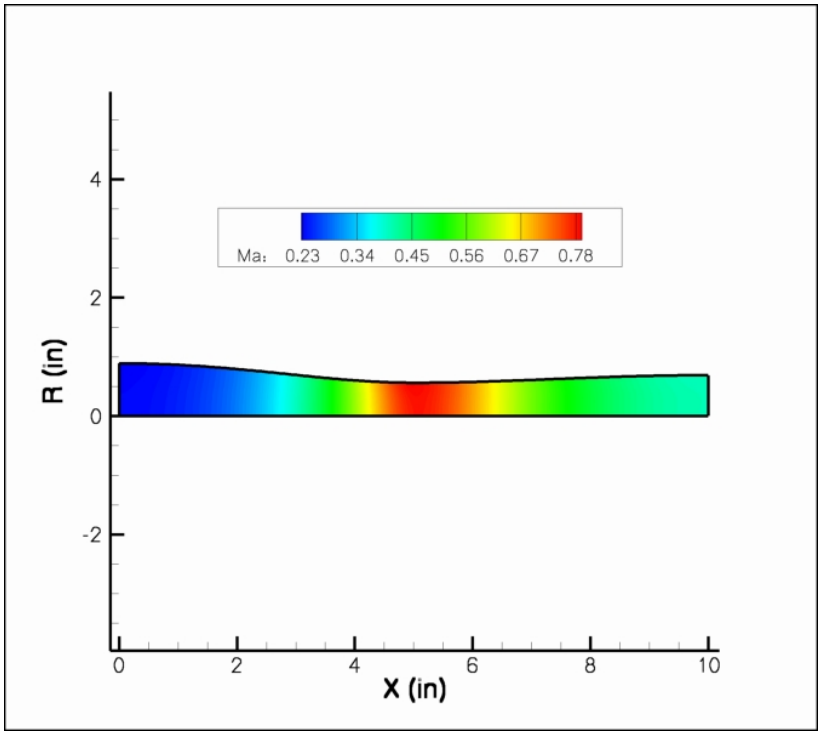


**Figure 11:** Steady State Mach Number Contour Plot for Subsonic Case

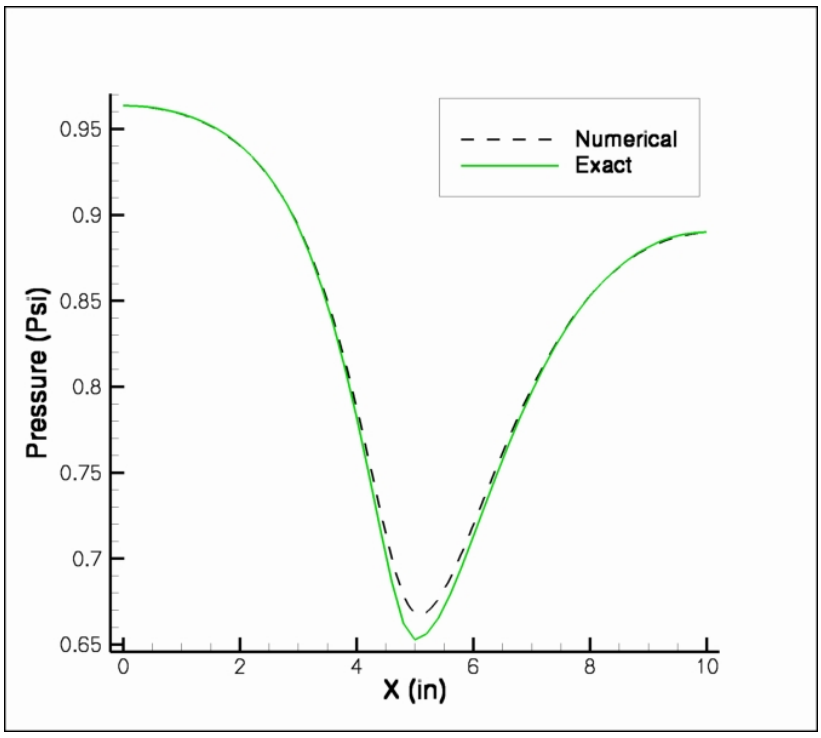


**Figure 12:** Centerline Pressure compared to Exact 1-D Solution for Subsonic Case

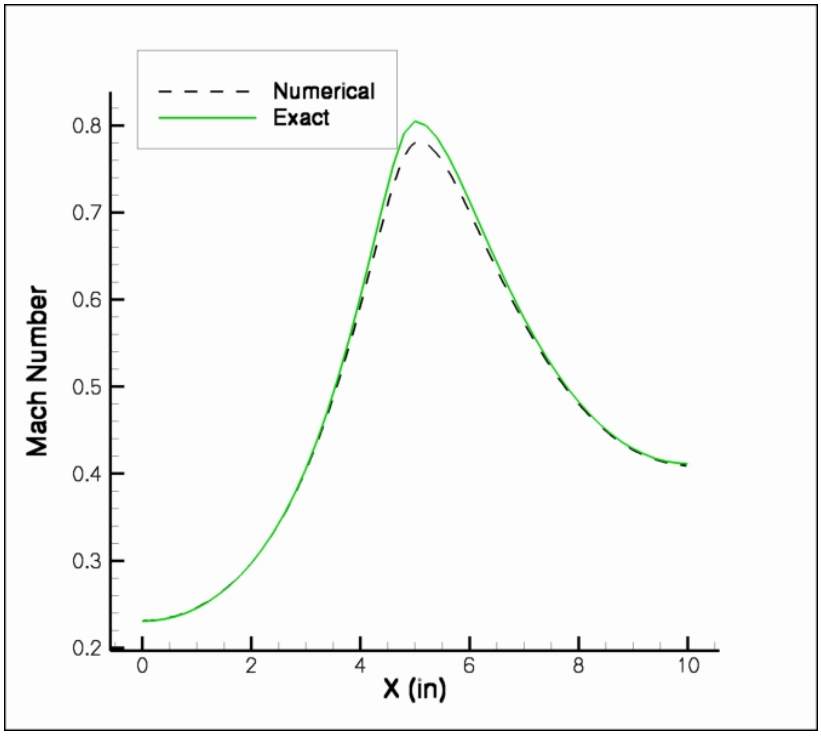


**Figure 13:** Centerline Mach Number compared to Exact 1-D Solution for Subsonic Case

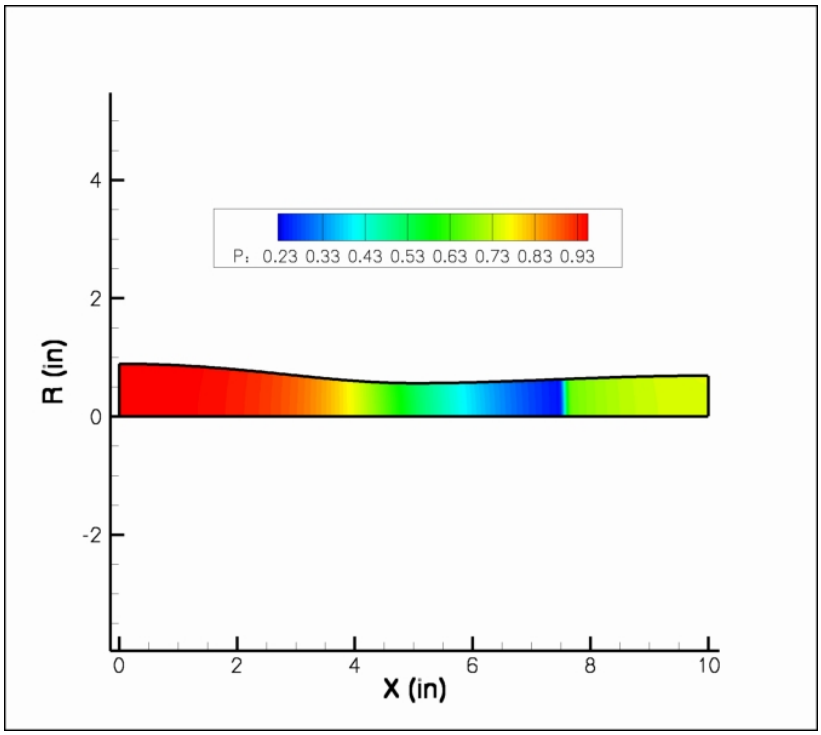


**Figure 14:** Pressure Contour for Overexpanded Case

dissipation used for stability. The pressure throughout the nozzle contains curvature. This curvature triggers an increase in artificial dissipation which tends to slow the flow and cause the under prediction of the Mach number.

### 5.2.3 Supersonic Flow with Shock in Diffuser

For this case, the flow enters the nozzle subsonically and transitions to sonic at the throat of the nozzle. As the flow continues to expand, the pressure falls below that of the outlet and a standing shock is formed to match the outlet pressure. The pressure and Mach contours for this flow case are found in Figure 14 and 15.

These plots clearly show the expected behavior. The standing shock in the diffuser is resolved nicely and is smeared only over relatively few cells. This behavior can be further seen with the one dimensional solution.

For this case, the centerline data is again compared with the analytical solution. This comparison is favorable. The only deviation from the analytical solution occurs near the shock. For the numerical solution, the shock appears to form approximately one cell downstream than predicted by the analytical solution. Additionally, there is a small undershoot

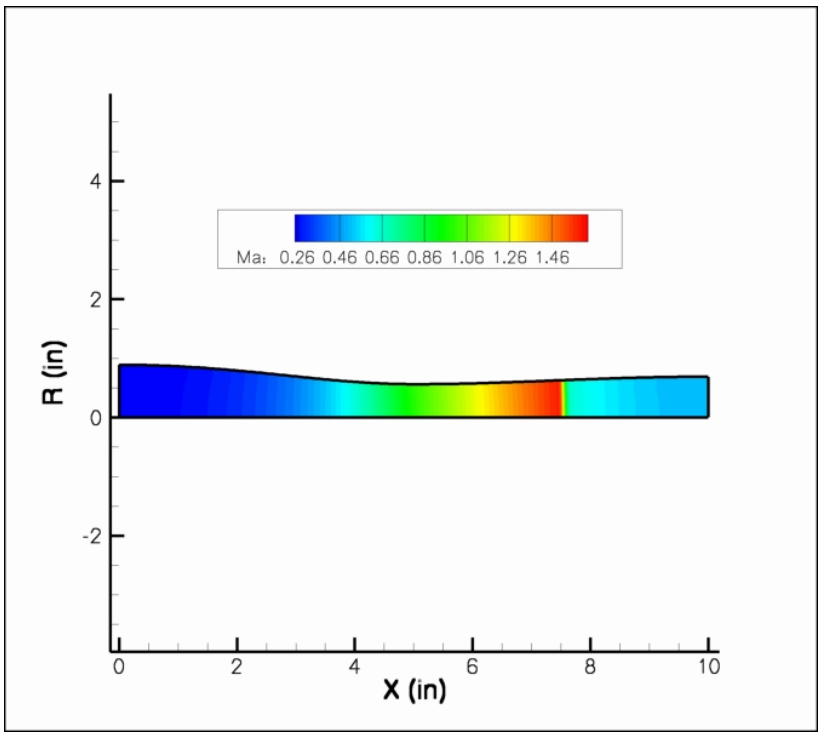


**Figure 15:** Mach Number Contour for Overexpanded Case

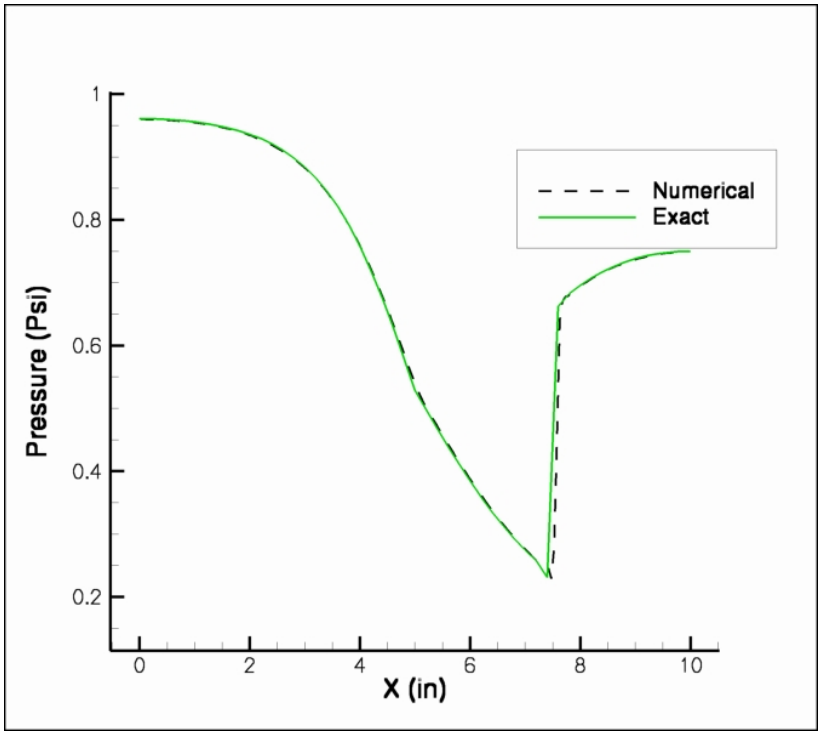


**Figure 16:** Centerline Pressure compared to Exact 1-D Solution for Overexpanded Case

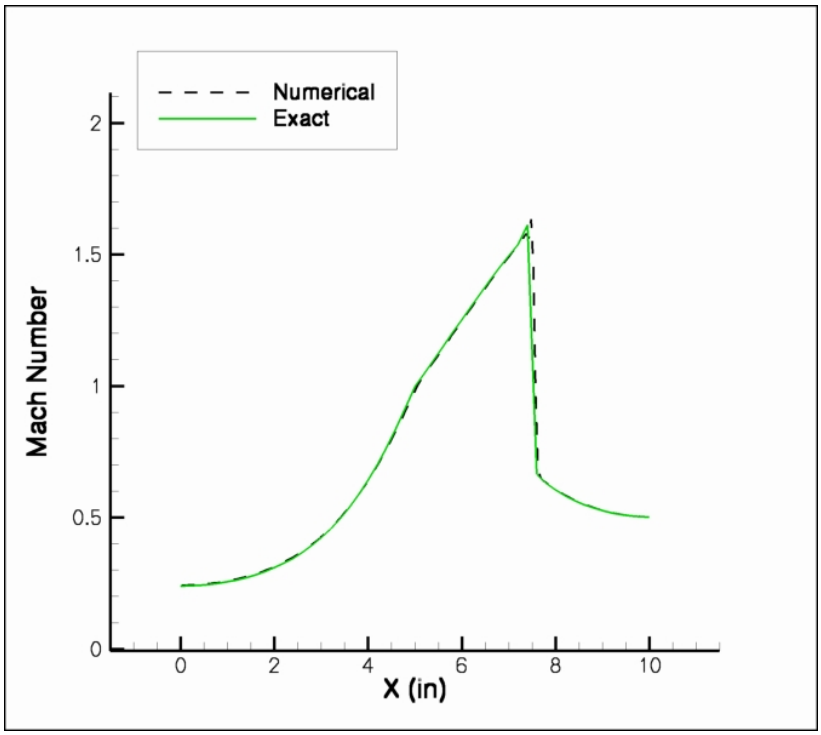


**Figure 17:** Centerline Mach Number compared to Exact 1-D Solution for Overexpanded Case

in pressure present just prior to the shock.

### 5.2.4 Supersonic Case

For the supersonic case, the flow is serverly underexpanded. As such, the flow accelerates throughout the length of the nozzle. The pressure and Mach countours for this case are presented in Figures 18 and 19. The one dimensional data is compared to the analytical solution in Figures 20 and 21. As these plots show, the centerline data agrees with the analytical solution with no places of particularly large discrepancy. Due to this close agreement with the analytical solution, the flow case is further examined in the spatial convergence study.

### 5.2.5 Spatial Convergence Study

In order to determine the spatial order of accuracy of the finite volume PCICE scheme, a spatial convergence study was performed using the supersonic converging-diverging nozzle. The spatial convergence study was performed in a similar manner to the temporal convergence study. The solution was found using a relatively large cell spacing. The cell spacing

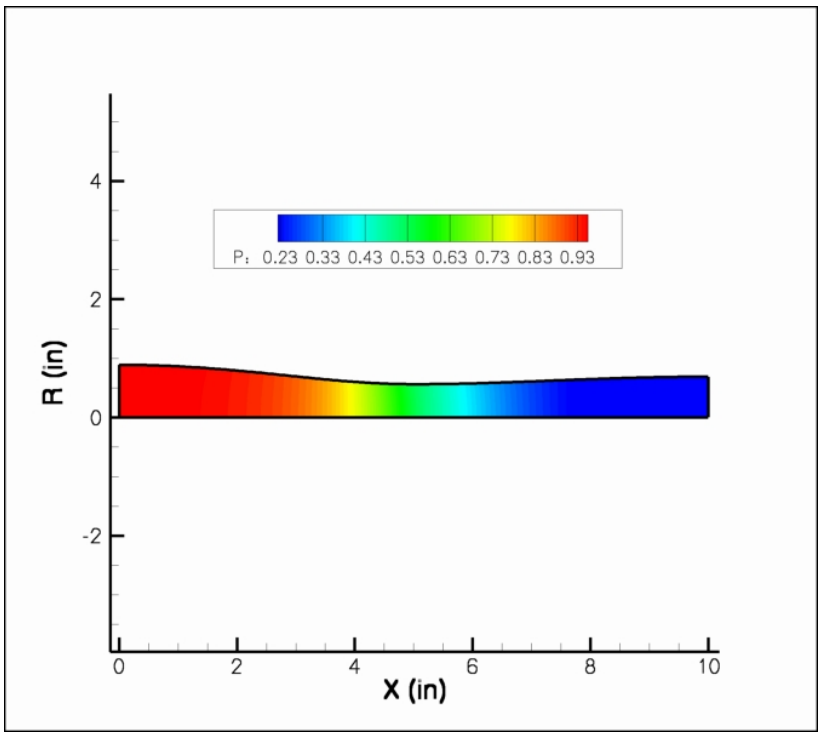


**Figure 18:** Pressure Contour for Supersonic flow case

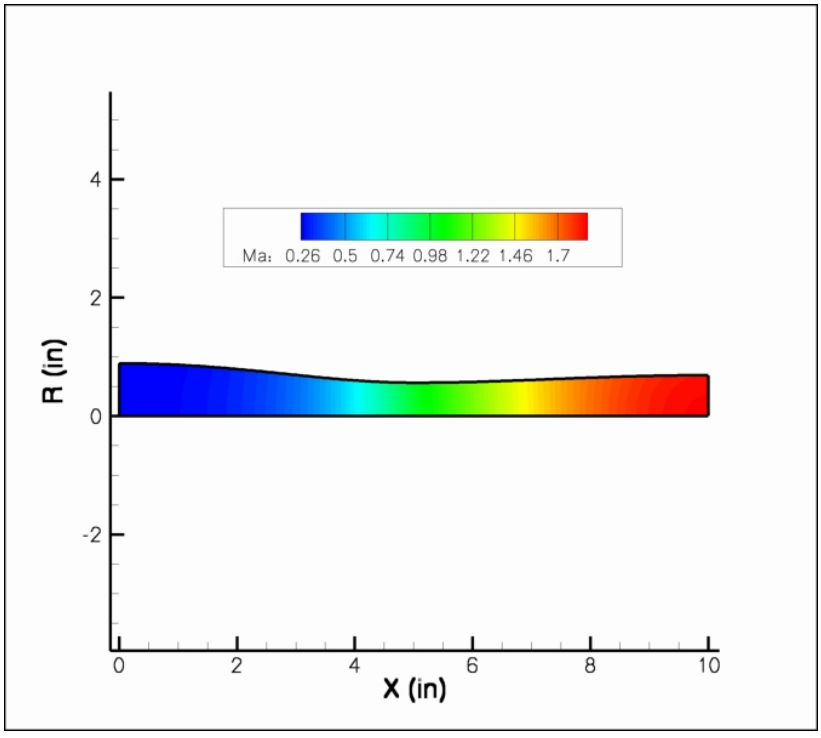


**Figure 19:** Mach Contour for Supersonic flow case

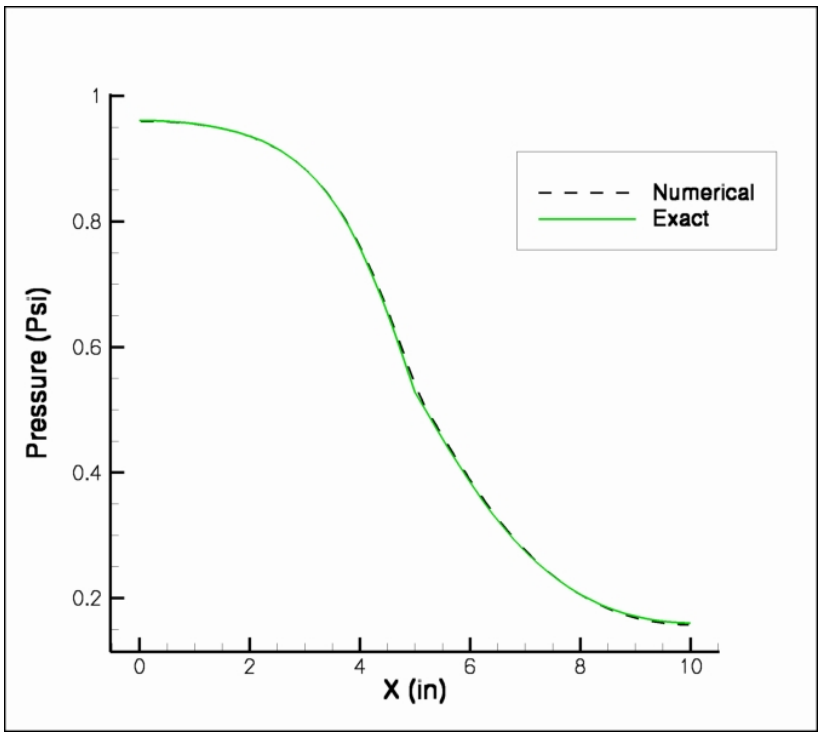


**Figure 20:** Centerline Pressure compared to Exact 1-D Solution for Supersonic Case

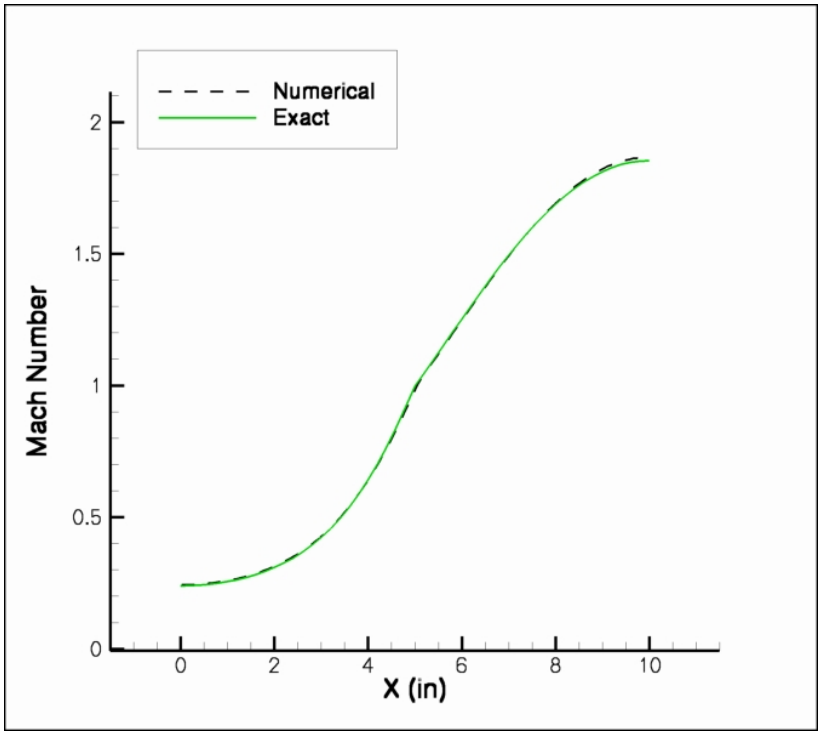


**Figure 21:** Centerline Mach Number compared to Exact 1-D Solution for Supersonic Case

**Table 2:** Results of Spatial Convergence Study

| Normalized Cell Spacing | f *psi* |
|---|---|
| 4.0 | 0.99940333 |
| 2.0 | 0.99981319 |
| 1.0 | 0.99992713 |
| 0.0 | 0.99997101 |
| $p = 1.84681096$ | |

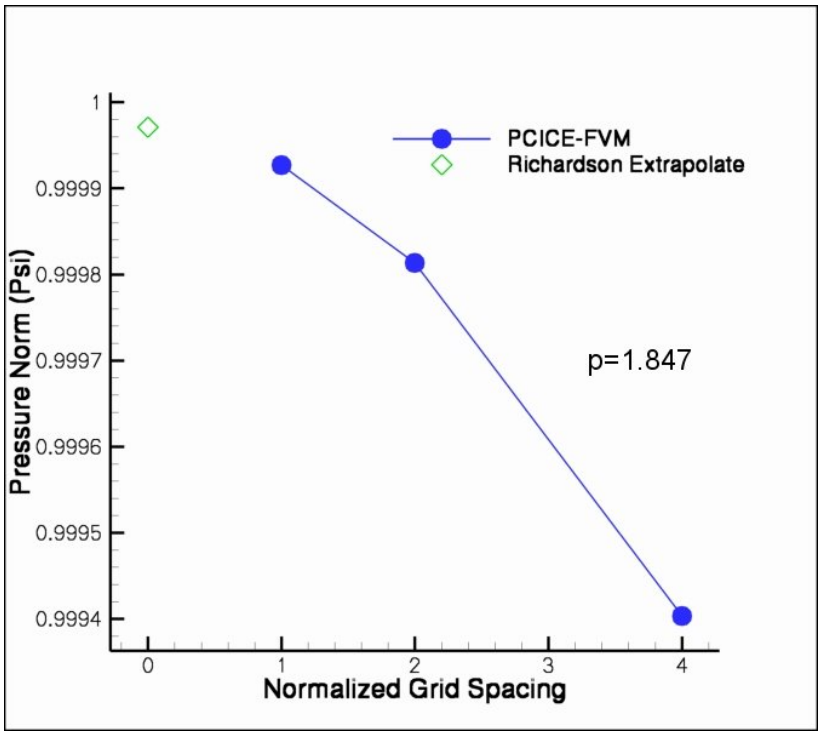


**Figure 22:** Result of Grid Convergence Study

was subsequently cut in half and the problem was solved again. For each cell spacing, the same norm calculated in Equation 198 was recorded.

For this problem, the three cell spacings used corresponded to 0.04, 0.02, and 0.01 inches. The problems were solved using the same CFL safety factor and smoothing parameters as before. In order to better distinguish between the solutions of the three mesh spacings, the steady state tolerance was reduced from $10^{-6}$ to $10^{-10}$. The results of the spatial convergence study can be found in Table 5.2.5. Included with these results is the Richardson extrapolate. For this flow case, the total pressure should be conserved throughout the nozzle, giving a theoretical value of 1.0 psi for the norm.

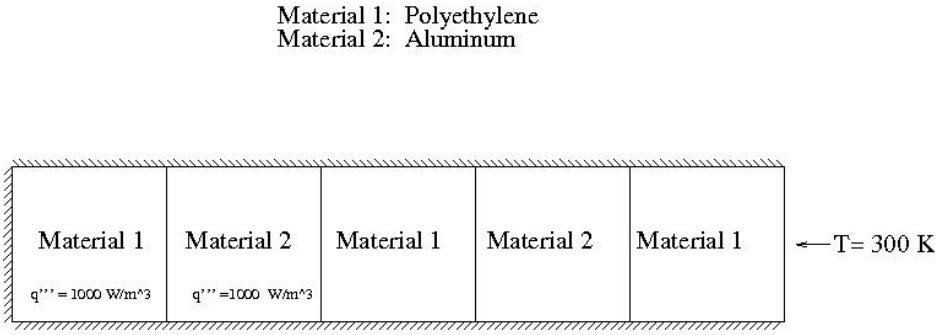


**Figure 23:** Conduction Benchmark Problem Setup

**Table 3:** Physical Parameters of Materials used in Conduction benchmark

| | Material 1 | Material 2 |
|---|---|---|
| $\rho$ | $1020\frac{kg}{m^3}$ | $2700\frac{kg}{m^3}$ |
| $c_v$ | $1960\frac{J}{kg-K}$ | $900\frac{J}{kg-K}$ |
| $k$ | $0.128\frac{W}{m-K}$ | $210\frac{W}{m-K}$ |

The results to the spatial convergence study are plotted in Figure 22. The spatial discretization is approximately second order. In practice, the theoretical order of convergence is rarely achieved. The global order of convergence is reduced by things such as boundary conditions, artificial dissipation, grid nonorthogonality, and the presence of shocks [24]. This spatial accuracy compares well with the accuracy achieved in the finite element PCICE scheme, which was 1.898 [14].

### *5.3 Conduction Benchmark*

In order to verify the treatment of conduction terms and the incorporation of conjugate heat transfer, a solid phase heat transfer problem was constructed. This problem consists of five slabs of material, alternating between polyethylene and aluminum. The slabs are surrounded on three sides by adiabatic walls, while the fourth wall has a fixed temperature of 300 K. In the two left most slabs, a heat generation equal to 1,000 $W/m^3$ is present. Each slab is 0.1 m square. Figure 23 summarizes this information. The physical parameters used for the two materials present are summarized in Table 5.3.

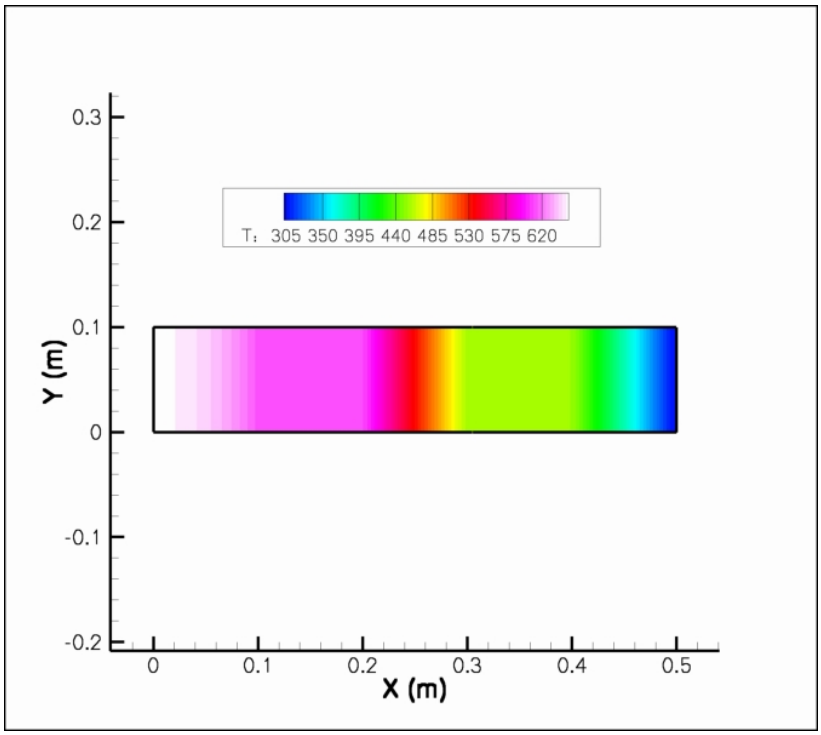


**Figure 24:** Steady State Temperature Distribution

This problem was chosen for several reasons, some of which have been mentioned previously. First, the spatial discretization used to approximate the conduction operator is tested. The addition of heat generation further tests this approximation by causing the temperature to have the form of a quadratic. Second, since the problem is insulated at the top and bottom and the material properties are uniform in this direction, the problem is completely one dimensional. By treating the thermal conductivity as a constant, the steady state analytical solution to this problem can be derived and compared to the numerical solution. Finally, since the variation in thermal conductivity is serval orders of magnitude at the interface between materials, the implementation of conjugate heat transfer can be confirmed. If the approximation for the heat flux at the interface did not match both boundary temperature and heat flux, the numerical solution would be inaccurate and would not compare well to the analytical solution.

Figure 24 shows the steady state temperature distribution within the materials. As this figure shows, the numerical solution is indeed one dimensional. The analytical solution to the problem was derived by treating the physical properties of the materials as constants and treating the heat generation as fixed. Through the use of the boundary conditions and

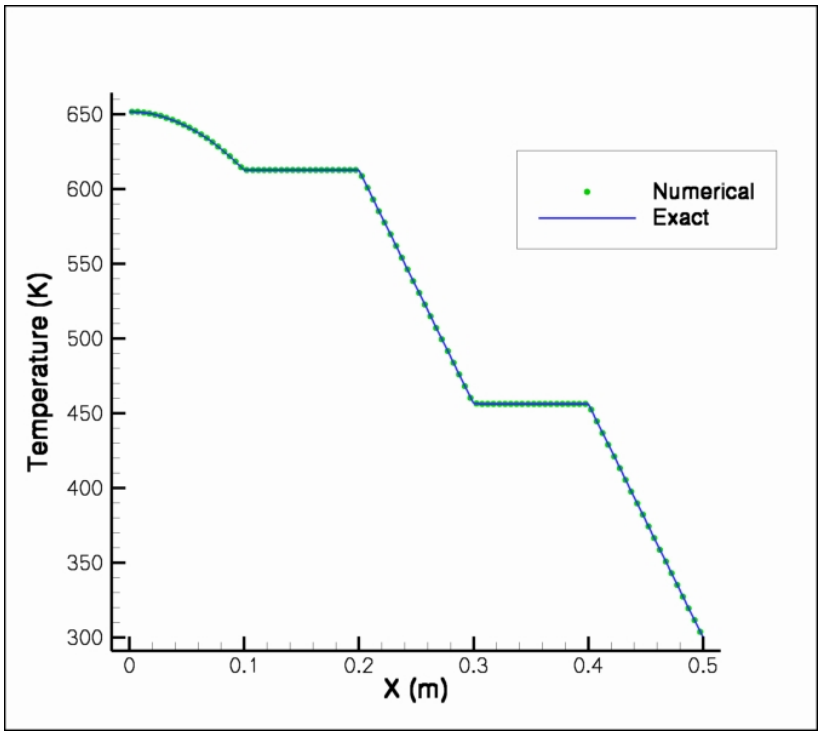


**Figure 25:** Numerical and Exact Solution to Conduction benchmark problem.

analytically enforcing the condition that temperature and heat flux at the interface match, an analytical solution for the temperature profile thoughout the whole domain can be found. Figure 25 contains the analytical solution plotted along with the numerical solution of the problem.

The analytical solution found in Figure 25 is found by solving the steady state heat equation for each slab. The equation for each slab is given by:

$$k_i \frac{dT}{dx} = -q_i \tag{204}$$

By integrating this equation with respect to x and enforcing continuity of temperature and heat flux at the interface between materials, the steady state temperature distribution can be given by the following equation.

**Table 4:** Physical Parameters of Materials used in Conduction benchmark

| Region | A | B | C |
|---|---|---|---|
| 1 | -3,906.250 | - | 651.729 |
| 2 | -2.381 | -0.476 | 612.667 |
| 3 | - | -1,562.500 | 612.595 |
| 4 | - | -0.952 | 456.345 |
| 5 | - | -1,562.500 | 456.250 |

$$T(x) = \begin{cases} A_1 x^2 + C_1, & 0 \leq x < 0.1; \\ A_2 x^2 + B_2 x + C_2, & 0.1 \leq x < 0.2; \\ B_3 x + C_3, & 0.2 \leq x < 0.3; \\ B_4 x + C_4, & 0.3 \leq x < 0.4; \\ B_4 x + C_4, & 0.4 \leq x \geq 0.5. \end{cases} \tag{205}$$

where the constants for each region are given in Table 5.3.

As Figure 25 shows, the numerical solution is nearly indistinguishable from the exact solution. Figure 26 contains the relative error between the numerical solution and exact solution plotted as a function of x. The maximum error is seen in the first material and is equal to 0.008 percent. This maximum error occurs at $x = 0.1m$, the first interface between the two materials. Throughout the rest of the domain, the error is essentially zero. This occurs because the analytical temperature distribution is essentially linear in these parts of the domain and the approximation of the heat flux assumes a linear dependence in temperature. Therefore, the approximation can capture the exact temperature distribution to a high level accuracy. In the first region, where the temperature distribution is nonlinear, the accuracy of the numerical scheme begins to decline. The error encountered in this region is most likely due to the spatial discretization error.

### *5.4 Channel Flow over Fuel Region*

In this problem, a solid section of fuel material is cooled by helium flowing on each side. This problem was selected for three reasons. First, the performance of the PCICE-FVM scheme for viscous flow can be demonstrated and examined. Second, and more importantly,

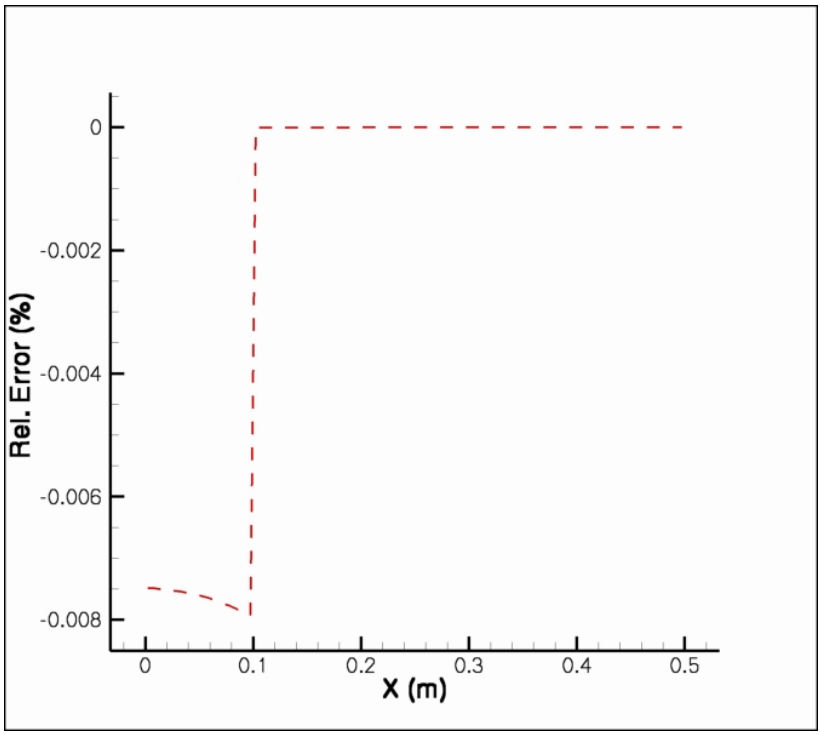


**Figure 26:** Percentage Error between Numerical and Analytical Solution

the conjugate heat transfer formulation is tested within a fluid/solid system. Finally, a loose coupling with the neutron transport code EVENT can be explored.

### 5.4.1 Problem Description

The geometry of this problem consists of a rectangular slab of fuel material which is 0.2 m wide and 1 m high. The solid itself is represented by a single homogenized material. The properties of the material correspond to the homogenized properties of a pebble bed reactor and consists of approximately 30 percent fuel and 40 percent graphite. The remaining 30 percent is given by gaps and coolant. The fuel itself is Uranium oxide. In order to reduce the complexity of the problem, the material is represented as a single solid phase and is given reasonable material properties. These material properties can be found in Table 5.4.1. On each side of the solid is a channel of helium which is 0.2 m in width. The helium is at 5 MPa and 750 K. The mass flow rate through the channel is held constant with an inlet velocity of 20 m/s. The total power generated within the solid was set to 200 kW. The spatial distribution of this power was determined by calculating the fission reaction rate at various points within the solid and normalizing it to the specified value. The problem setup

**Table 5:** Physical Parameters for Homogenized Fuel Material

| | |
|---|---|
| $\rho$ | $4191 \frac{kg}{m^3}$ |
| $c_v$ | $700 \frac{J}{kg-K}$ |
| $k$ | $24 \frac{W}{m-K}$ |

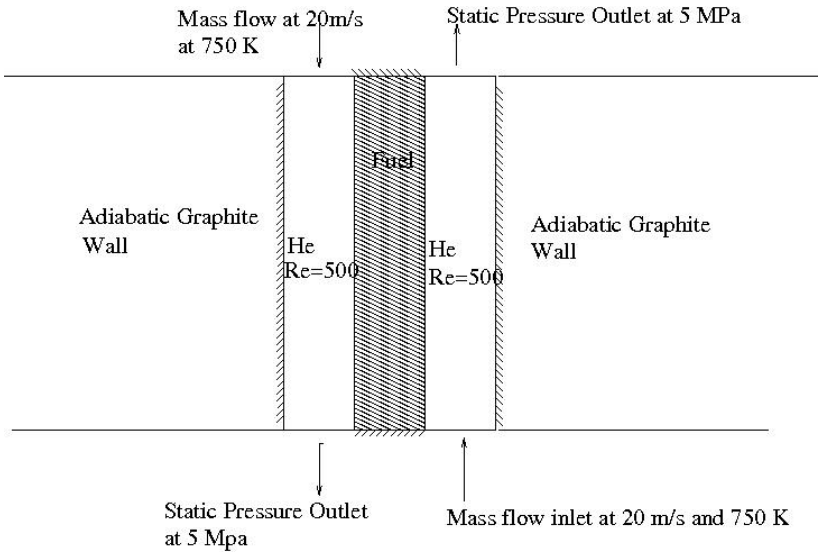


**Figure 27:** Geometry of Simplified Fuel-Fluid System

is given in Figure 27.

### 5.4.2 Numerical Results

Before the fluid flow and conduction is simulated, the heat generation in the solid is required. To determine the spatial distribution of this heat source, EVENT was used to calculate the fission reaction rate. The cross-sections for the solid were chosen to conform to the homogenized pebble bed reactor. In order to give the flux a more interesting profile, the solid was divided into four regions. The lower left and upper right sections were given cross-sections evaluated at 1600 K. The other two regions were given cross-sections evaluated at 600 K. At the upper and lower boundary of the solid, reflective boundary conditions are applied. At the left and right boundary of the domain, albedo boundary conditions were used. The coefficient of this boundary condition was adjusted such that the system was

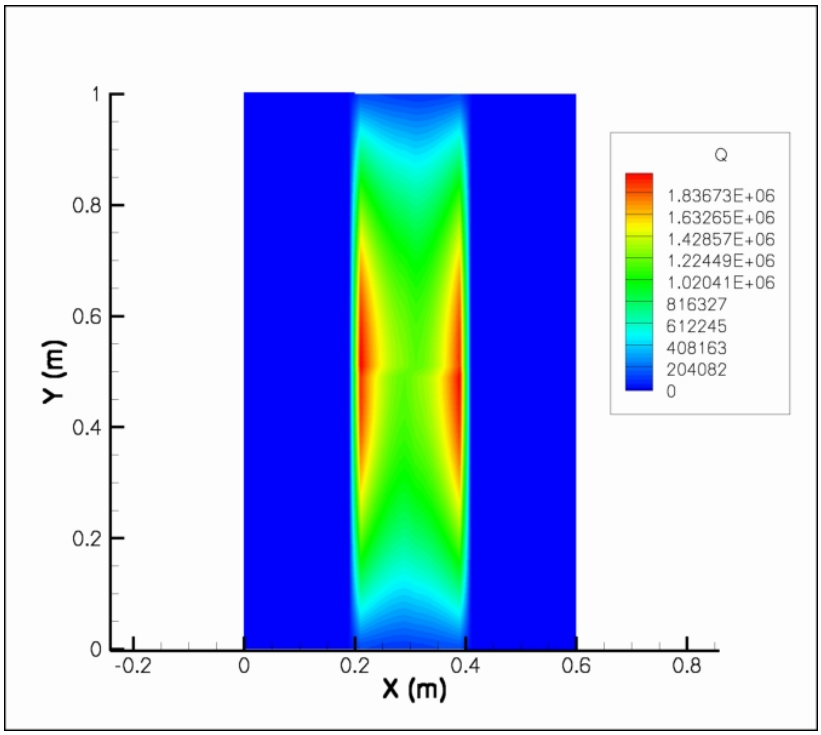


**Figure 28:** Volumetric Heat generation within the domain

nearly critical. In addition to adjusting the albedo boundary condition, additional slabs of graphite were added to increase the amount of moderation in the system. This additional graphite was not included within the fluid simulation. The mesh used for EVENT consisted of uniform, quadrilaterals with a side length of 1 cm. With the problem setup, the scalar flux throughout the domain was calculated and the fission reaction rate for each cell was calculated. The reaction rate was then written to a data file that could be read into the fluids solver [22]. Figure 28 shows the volumetric heat generation within solid.

With the power distribution within the solid determined, the solid conduction and fluid flow within the domain can be solved. For the fluid/solid domain, adiabatic walls were placed at the upper and lower boundary of the solid. On each side of the channel, a no-slip boundary condition is applied. On the outer boundary of the each channel, an adiabatic boundary condition is again applied. At the inlet of each channel, a specified mass flux boundary condition is imposed. This mass flow corresponded to 12.8 kg/s. Static pressure outlets at 5 MPa were used for the outlet boundary condition.

The viscosity of the fluid was set such that the Reynolds number within the channel was 500. For the inlet velocity of 20 m/s, this gives a viscosity of 0.0256 Pa-s. Using the Prandtl

number for Helium, 0.654 at 750 K, the thermal conductivity was equal to 203.32 W/m-K. Clearly, these properties do not reflect the physical properties held by helium. However, in order to accurately demonstrate the workings of the conjugate heat transfer formulation, it was decided that the Reynolds number should be low to ensure that a boundary layer adequately developed and a significant amount of heat was transfered to the fluid. In order to accelerate the convergence to steady state, the density and specific heat of the solid were set to low values. This action gives the solid a low thermal capacitance, causing it to heat up quickly. Once the solid temperature reached a steady state, the density and specific heat were set to the appropriate values. The mesh within this problem was chosen to conform to the mesh used in the transport calculations. As such, a uniform mesh of quadrilaterals with side of 1 cm was used. As far as the solver parameters were concerned, the CFL safety factor was set to 0.1 and the smoothing parameters were set to 0.1 and 0.0002 for the second order and fourth order terms respectively.

The velocity and pressure contours within the domain can be seen in Figures 29 and 30, respectively. As Figure 29 shows, the velocity enters each channel at 20 m/s. As the boundary layer grows, the centerline velocity is increased, while the velocity at the wall is near zero. It should be noted that the velocity in the plot does not go exactly to zero at the wall. This occurs because the visualization program (Tecplot 10.1) has extrapolated the cell centered data to the boundary of the domain. This extrapolation appears to be low order and causes the velocity at the boundary to be non-zero. The pressure contour within the domain shows a smooth pressure drop from the inlet to the outlet of the each channel. Additionally, the pressure within the solid is shown to be uniform. This behavior occurs because the solid pressure is initialized to the ambient pressure and never updated. Figure 31 shows the temperature contour within the domain. In the solid, the temperature is peaked at the center. The temperature then decreases towards the upper and lower boundaries. This peaking is mostly associated with the spatial distribution of the heat source. Toward the fluid/solid interface, the temperature in the solid also decreases due to cooling by the helium. From this plot, it is not possible to see a thermal boundary layer within the fluid because of the wide range of temperatures displayed. For this problem,

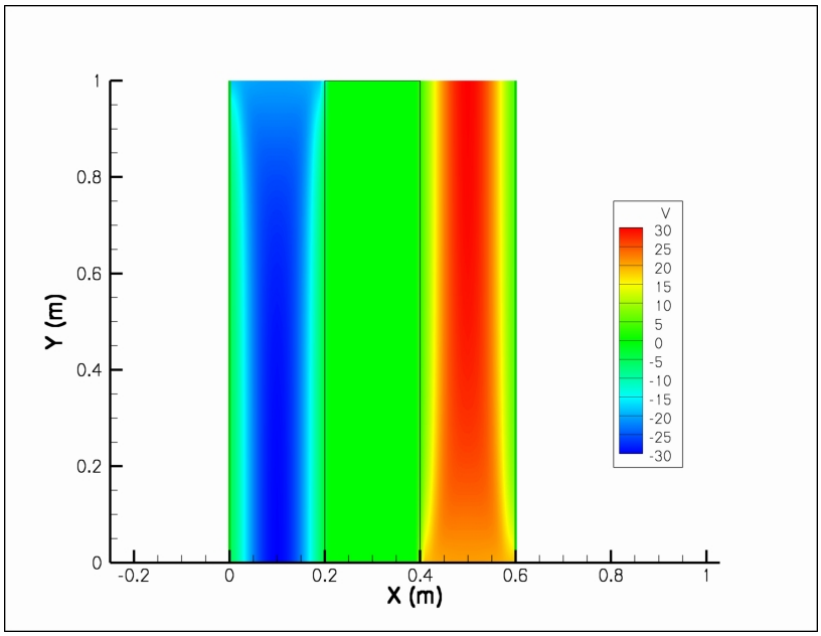


**Figure 29:** Contour Plot of Velocity

the fluid temperature at the wall was raised by approximately 15 K. Figure 32 shows the temperature contour of the domain over this narrow range. This plot clearly shows the development of a thermal boundary layer as the fluid is heated by the solid. The results displayed here have neglected the contribution due to viscous heating. This contribution was left out to allow for a global energy balance.

### 5.4.3 Validation

Since this is a complex problem, analytical solutions are not available. However, to confirm the results of the simulation, several global parameters can be validated to ensure the accuracy of the formulation. In particular, the mass balance and energy balance within the entire system will be checked.

In order for the numerical solution to conserve mass, the mass entering the domain must be equal to the mass exiting the domain at steady state. This requirement can be written as the integral over the boundary of the mass flux ($\rho\vec{u}$).

$$\int_{\Gamma} \rho\vec{u} \cdot \vec{n} d\Gamma = 0 \tag{206}$$

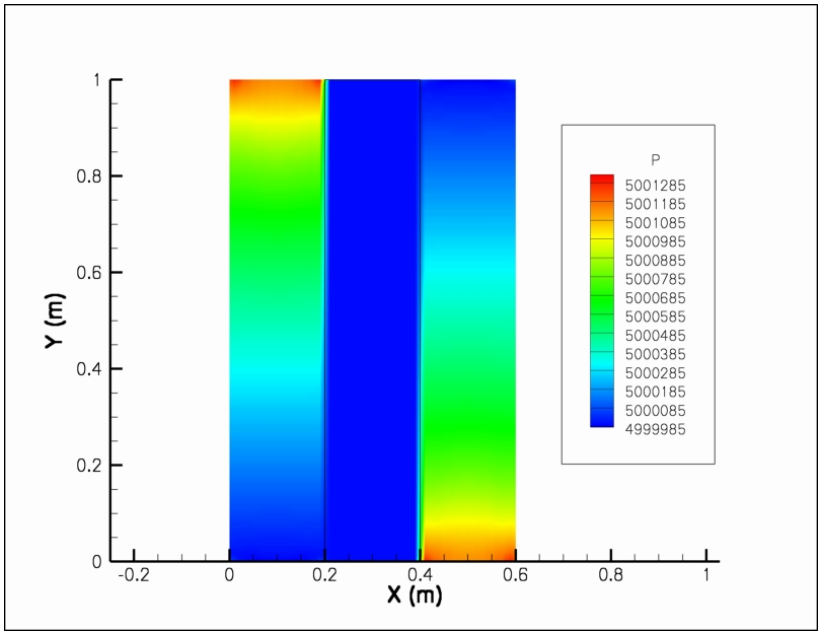


**Figure 30:** Contour Plot of Pressure

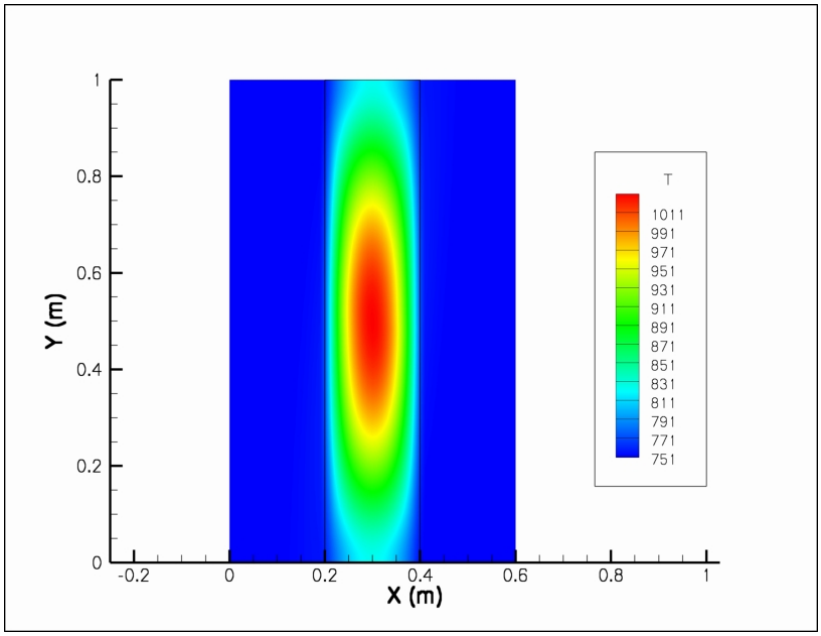


**Figure 31:** Contour Plot of Temperature

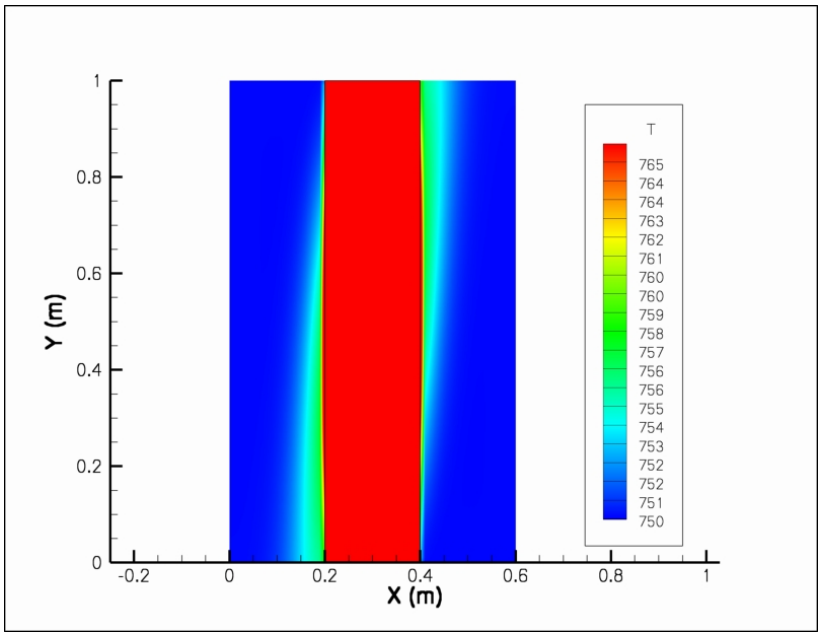


**Figure 32:** Contour Plot of Temperature within the Fluid

In discretized form, this requirement can be expressed as the sum over the boundary faces of the mass flux normal to that face. In reality, the above requirement will not be exactly zero. This deviation from zero is the mass error. The expression used to calculate the mass error is given by:

$$\varepsilon = \sum_{k \epsilon \Gamma} \rho u dy_k - \rho v dx_k \tag{207}$$

For the energy balance, the energy exiting the domain must be balanced by the energy entering the domain and the energy generated within the domain once the system has reached steady state. Ignoring conduction occurring at the outer boundary of the domain, this requirement can be represented mathematically by:

$$\int_\Gamma \rho \vec{u} h_t \cdot \vec{n} d\Gamma = \int_V q''' dV \tag{208}$$

where $q'''$ is the volumetric heat generation (previously represented by $i$) and $h_t$ is the total enthalpy. In descretized form, this requirement can be used to define the energy deficit ($\delta$) and is calculated using the following equation.

$$\delta = \sum_{k \epsilon \Gamma} \rho u h_t dy_k - \rho v h_t dx_k - \sum_i q_i''' \tag{209}$$

With these definitions, the mass error and energy deficit within the numerical solution is calculated. The results of these calculations can be found in Table 5.4.3. The mass error in the solution was found to be $2.68x10^{-8} \quad kg/s$. This value corresponds to $1.05x10^{-7}\%$ of the specified inlet mass flow. Clearly, the numerical solution properly conserves the mass of the system. This result should be expected as the formulation used for the numerical solution solves the conservative form of the governing equations. Additionally, an integral method is used to spatially discretize the domain. These two factors ensure that mass conservation is enforced for each cell for every time step. The net effect of this treatment is that mass is well conserved for the entire domain.

For the global energy balance, the energy deficit was found to be approximately $-320W$. On the surface, this appears to be a large energy deficit; however, this value represents only 0.16% of the total power generated within the system. This percentage error is well within that expected by any numerical solution. Additionally, accounting for the energy transfered by conduction at the inlet and outlet could drive the energy deficit further toward zero. The most important result of this energy balance is that the proper amount of power is transfered from the solid to the fluid. The low relative error seen in the energy balance confirms that the conjugate heat transfer implemented within this code properly couples the solid phase conduction to the fluid flow within the system. This proper coupling should not be surprising for a steady state problem, as previous forms of CHT have properly achieved this. What separates this result from previous CHT formulations is that no special consideration was given to the interface between the fluid and solid. Additionally, no special actions were required to ensure that the heat flux between the fluid and solid was continuous.

**Table 6:** Results of Mass and Energy Balance

| | Mass (kg/s): | Energy (W): |
|---|---|---|
| In: | 25.68 | 100,028,880 |
| Out: | 25.68 | 100,228,561 |
| Generated: | 0 | 200,000 |
| Net: | $2.68x10^{-8}$ | -319.27 |
| Error (%): | $1.05x10^{-7}$ | 0.16 |

# CHAPTER VI

# CONCLUSIONS AND FURTHER WORK

For this research, a two dimensional, finite volume formulation of the PCICE algorithm was created. This formulation was based on cell-centered finite volumes and currently employs quadrilaterals. The spatial differencing was based upon linear interpolation and Jameson's form of artificial dissipation was used for stability and shock capturing. This formulation was created for the Cartesian form of the Navier-Stokes equations and the axisymmetric form of the Euler equations. The formulation was found to be consistent with the governing equations. The spatial order of convergence was found to be just under two and the temporal convergence was second order.

In addition to solving fluid flow problems, conjugate heat transfer capabilities were included in the solver. This addition was carried out using a newly developed form of CHT, in which the interface heat flux is treated without the use of boundary conditions. With this treatment, continuity of temperature and heat flux was guaranteed. For the finite volume formulation, a simple two step approximation of the temperature gradient was implemented. This approximation allowed for large changes in thermal conductivity across a cell face and ensured that the continuity condition was met. This implementation was confirmed using a solid phase conduction benchmark. The formulation was further tested when crossflow over a solid fuel was examined. In both of these test problems, the formulation performed well and produced physically accurate results. Finally, a primitive form of coupling with the radiation transport code EVENT was carried out. This coupling was based upon file I/O and provided a heat generation source to the fluids/solid solver.

This work represents a positive initial step in developing a robust computational fluid dynamics solver which can be used in conjuction with a radiation transport code. To extend this work, several key additions must be made. First, approximations used to estimate face centered quantities should be refined to allow for a greater variety of meshes. Second, the

temporal treatment of the diffusive terms should be improved. Ideally, a fully implicit treatment would be implemented. This treatment is currently utilized in the finite element form of the PCICE algorithm, but has yet to be examined for finite volumes [18]. Finally, the solver itself should be embedded into EVENT or any suitable radiation transport code. By fully imbedding the solver, information exchange between the radiation transport solver and the fluids solver would be simplified and fully coupled simulations of reactors could be performed.

# REFERENCES


[1] Ahmadi, M. and Ghaly, W. S., "A finite volume method for the two-dimensional euler equations with solution adaptation on unstructured meshes," 1987.

[2] Berry, R., "Notes on the pcice method: Simplification, generalization, and compressibility properties," *Journal of Computational Physics*, no. 215, pp. 6–11, 2006.

[3] Berry, R. A. Class Notes, Summer 2006.

[4] Chung, T. J., *Computational Fluid Dynamics*. Cambridge, United Kingdom: Cambridge University Press, 2002.

[5] Dick, E., "Introduction to finite volume techniques in computational fluid dynamics," *VKI An Introduction to Computational Fluid Dynamics*, 1989.

[6] Fedorov, A. and Viskanta, R., "Three-dimensional conjugate heat transfer in the microchannel heat sink for electronic packaging," *International Journal of Heat and Mass Transfer*, vol. 43, pp. 399–415(17), February 2000.

[7] Ferziger, J. H. and Peric, M., *Computational Methods for Fluid Dynamics*. Berlin: Springer, 2nd. edition ed., 1997.

[8] Ha, M. Y. and Jung, M. J., "A numerical study on three-dimensional conjugate heat transfer of natural convection and conduction in a differentially heated cubic enclosure with a heat-generating cubic conducting body," *International Journal of Heat and Mass Transfer*, vol. 43, pp. 4229–4248, February 2000.

[9] Haarmann, T. M. and Koschel, W. W., "Compuation of wall heat fluxes in cryogenic h2o2 rocket comustion chambers," in *38th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, (Indianapolis, IN), July 2002.

[10] Hirsch, C., *Numerical Computation of Internal and External Flows*. New York, NY: John Wiley & Sons, 1990.

[11] Hoffmann, K. A. and Chiang, S. T., *Computational Fluid Dynamics*. Wichita, KS: Engineering Education System, 2nd. edition ed., 2000.

[12] Incropera, F. P. and DeWitt, D. P., *Fundamentals of Heat and Mass Transfer*. New York: John Wiley and Sons, 5th. edition ed., 2002.

[13] Jameson, A., Schmidt, W., and Turkel, E., "Numerical solution of the euler equations by finite volume methods using runge-kutta time-stepping schemes," in *Fluid and Plasma Dynamics Conference, 14th.*, (Palo Alto, CA), June 1981.

[14] Martineau, R., "The pcice-fem scheme for highly compressible axisymmetric flows," *Computers and Fluids*, vol. 36, no. 7, pp. 1259–1272, 2007.

[15] Martineau, R. and Berry, R., "The pressure-corrected ice finite-element method (pcice-fem) for compressible flows on unstructured meshes," *Journal of Computational Physics*, pp. 659–685, March 2004.

[16] Martineau, R. and Berry, R., "Characteristic boundary conditions for the two-step taylor-galerkin fem," *Computer Methods in Applied Mechanics and Engineering*, vol. 195, no. 7, pp. 742–762, 2006.

[17] Martineau, R. C. Written and Oral Correspondance, June 2007.

[18] Martineau, R. C., "The pcice algorithm coupled with implicit conjugate heat transfer (cht) for reactor simulations." Conference Presentation, April 2007.

[19] Mavriplis, D. J., *Solution of the Two-Dimensional Euler Equations on Unstructured Triangular Meshes*. PhD thesis, Princeton University, 1987.

[20] Munson, B. R., Young, D. F., and Okiishi, T. H., *Fundamentals of Fluid Mechanics*. New York: John Wiley and Sons, 4th. edition ed., 2002.

[21] Nithiarasu, P., Zienkiewicz, O. C., Sai, B. V. K. S., Morgan, K., Codina, R., and Vázquez, M., "Shock capturing viscosities for the general fluid mechanics algorithm," *International Journal for Numerical Methods in Fluids*, vol. 28, pp. 1325–1353, dec 1998.

[22] Park, R. Written and Oral Correspondance, June 2007.

[23] Rahaim, C. P., Kassab, A. J., and Cavalleri, R. J., "Coupled dual reciprocity boundary element/finite volume method for transient conjugate heat transfer," *Journal of Thermophysics and Heat Transfer*, vol. 14, no. 1, pp. 27–38, 2000.

[24] Roache, P., *Verification and Validation in Computational Science and Engineering*. Hermosa Publishers, 1998.

[25] Slater, J. W., "Nparc alliance cfd verification and valiadation archive," tech. rep., NASA Glenn Research Center, June 2007.

[26] Stein, C. F., Johansson, P., Bergh, J., Lofdahl, L., Sen, M., and el Hak, M. G., "An analytical asymptotic solution to a conjugate heat transfer problem," *International Journal of Heat and Mass Transfer*, vol. 45, pp. 2485–2500, November 2002.

[27] Webster, R. S., *A Numerical Study of the Conjugate Conduction-Convection Heat Transfer Problem*. PhD thesis, Mississippi State University, 2001.